كتاب اجاكس الجزء الاول

اقدم لك عزيزي القارئ اول كتاب عربي يتحدث عن تقنية اجاكس اسال الله ان ينفع به امة الاسلام
اللهم علمنا ما ينفعنا و انفعنا بما علمتنا و زدنا علما انك انت العليم الحكيم

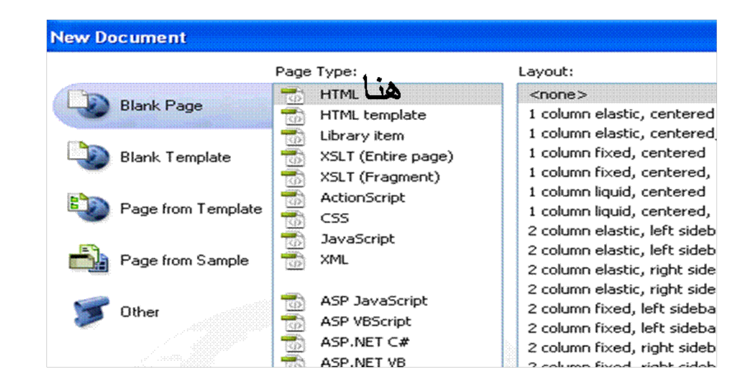
اخوكم ايمن كزة

سنبدأ من تحت الصفر و نصل بك الى مبرمج اجاكس محترف بمشيئة الله لا اريد منك الا ان تعرف القليل عن لغة HTML لاني ساشرح لك ما يلزمك من لغات ستلزمنا اثناء تطبيقات الاجاكس مثل(XML,CSS,PHP,javaScript)

لماذا اتعلم اجاكس؟
هناك عشرات المزايا الرائعة التي توفرها تقنية اجاكس ساكتفي بذكر مثال بسيط اذهب الان الى غوغل و اكتب كلمة دروس ستجد ان غوغل يحاول ان يحزر اذا كنت تبحث عنة دروس في الفوتوشوب و هذا ما يسمى الاكمال التلقائي

ما البرامج التي احتاجها؟

اروع برنامج هو الادوبي دريم ويفر بامكانك ان تطلبه من اي محل لبيع الدي في دي او تحميله من النت الان بعد ان نصبت البرنامج اختر من القائمة فايل الامر نيو File>new>HTML



وبعد ان اخترت اتش تي ام ال يفتح لك صفحة فيها كود جاهز كالصورة التالية

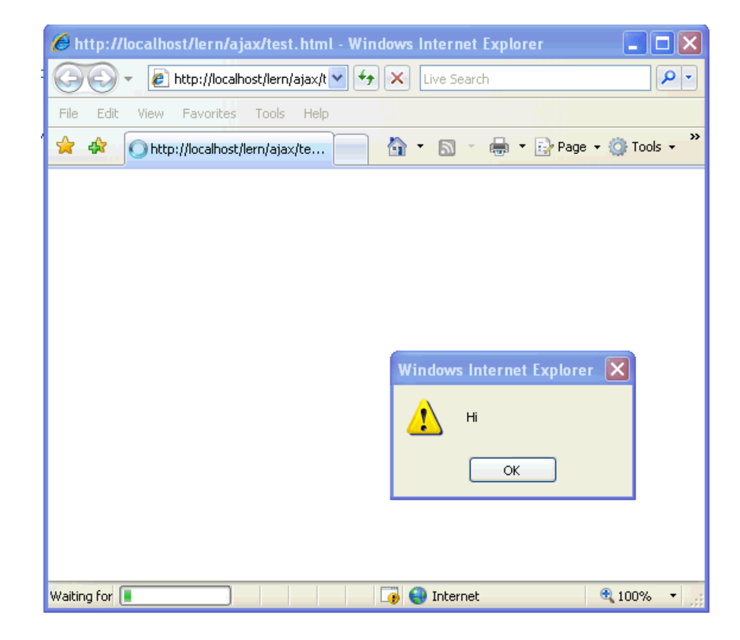
```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitiona
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; char
5  <title>Untitled Document</title>
6  </head>
7
8  <body>
9  </body>
10 </html>
11
```

الفصل الاول جافا سكريبت

هيا لننطلق مع اللغة اللذيذة جافا سكريبت

```
<html>
<head>
<script type="text/javascript">
alert("Hi ");
</script>
<title>My first script </title>
</head>
<body>
<h1> Ayman for php</h1>
</body>
</html>
```

و عند تشغيلك لهذا الكود يجب ان يعطي النتيجة التالية:



فقط السطر الغامق هو بلغة الجافا اما البقية فهي لغة الاتش تي ام ال اي فقط هذا السطر

```
alert("Hi ");
```

و لا حظ ايضا الفاصلة المنقوطة التي يجب ان تضعها في نهاية كل سطر **(;)** تعود ان تضع السيمي كولن لانك ان نسيتها بسكربت بي اتش بي سيفشل البرنامج كليا

هذا الكود الخفيف يعطيك نشاطا و يحمسك على تعلم هذه اللغة الرائعة فقط انسخه ثم احفظ الملف بأي اسم و شغله بالانترنت اكسبلورر مثلا .

هناك العديد من الكائنات الجاهزة في لغة الجافا سكريبت و لهذه الكائنات خواص و عليك ان تستدعي الخاصية و تعبئ قيمة بداخلها .

فمثلا الكائن اذا اردت ان تطبع جملة استخدم الكائن document

ثم استدعي الخاصية write()

ثم ضع القيمة التي تريدها

document.write("I love java");

سهل جدا اليس كذلك؟ طيب لناخذ مثالا اخرلنفرض انك تريد فتح نافذة جديدة لتكن غوغل مثلا و حدد كلمة البحث الا يبدو ذلك ممتعا؟

```
<script type="text/javascript">
window.open("http://google.com/search?q=islam","any_name","");
</script>
```

هذا الكود سيفتح نافذة جديدة و ياخذه الى غوغل و يبحث عن كلمة اسلام كل هذا دون علم المستخدم!!

document.title لتغيير عنوان الصفحة

document.bgcolor لتغيير خلفية الصفحة

location.hostname استخدمها لتعرف اسم شركة مزود الخدمة

navigator.appName مفيدة جدا لتحديد نوع المتصفح الذي يستخدمه الزائر

التعليقات

احيانا انت كمبرمج تضطر الى كتابة تعليقات تساعدك على تذكر الية عمل البرنامج او اي شيئ اخر تريد كتابته لكنه سيفسد الكود ان لم تستخدم (//) او (/*التعليق*/)

```
<html>
<head>
<script type="text/javascript">
// That was in 1/1/2010
alert("I love java");
/*
comment
annother comment
Hi ....
*/
</script>
<title>JavaScript</title>
</head>
<body>
<h1>Ayman for php </h1>
</body>
</html>
```

كما ترى فيمكنك ان تضع ما مئات الاسطر من التعليقات اذا استخدمت (/*عشرات الاسطر من التعليقات*/)

اما الدوبل سلاش فهي لتعليق سطر واحد // فهذه الجمل لن تظهر ابدا لانها مجرد تعليقات

تضمين ملف خارجي

احيانا تضع برنامج بحيث تستخدم ملف اسمه سكريبت1 اذا كان متصفح الزائر انترنت اكسبلور مثلا اما اذا كان متصفحه فايرفوكس فانك تريد استخدام الملف سكريبت2 على سبيل المثال

```
document.write("نص من ملف خارجي");
```

احفظ السطر السابق باسم script.js

ثم قم بتضمين الملف باستخدام خاصية src

```
<html>
<head>
<script type="text/javascript" src="script.js">
</script>
<title> برنامجي الاول </title>
</head>
<body>
</body>
</html>
```

احفظ الكود السابق باي اسم تريد مثلا Ajax1.html
سيعمل الكود ان شاء الله شرط ان تحفظ الملفين بنفس المجلد و ستظهر جملة نص من ملف خارجي مع اننا لم نكتبها بصفحتنا بل كتبناها بملف خارجي

معرفة نوع المتصفح

بامكان الجافا ان تكشف نوع المتصفح الذي يستخدمه الزائر و ذلك باستخدام الاوبجكت Navigator و هذا الاوبجكت له خواص فمثلا :

لمعرفة اسم المتصفح navigator.appName
لمعرفة رقم اصدار او الفيرجن navigator.appVersion
هذه الخاصية تعطيك تفاصيل اخرى navigator.userAgent

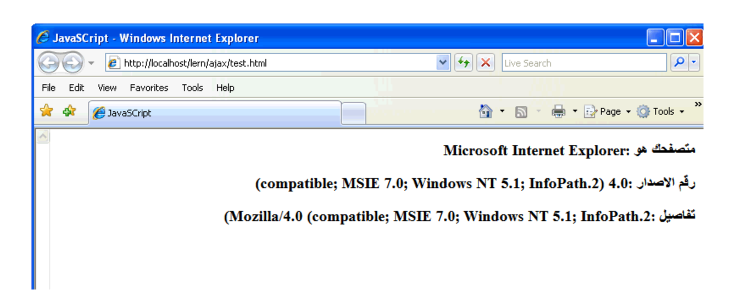
لنجرب مثال عملي

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
</head>
<body>
<script type="text/javascript">
document.write("<h3>متصفحك هو :"+navigator.appName+"</h3>");
document.write(" <h3>رقم الاصدار :"+navigator.appVersion+"</h3>");
document.write(" <h3>تفاصيل :"+navigator.userAgent+"</h3>");
</script>
</body>
</html>
```

لاحظ اشارة الزائد (+) ضرورية للجمع بين متغير و نص الان لنرى كيف سيعرض الفايرفوكس هذه الصفحة



بينما اذا جربت الانترنت الاكسبلورر سيعرضها هكذا



هذا مفيد جدا فيمكنك ان تستخدم السكربت المناسب لكل متصفح

الاحداث(Events)

عندما يضغط المستخدم على زر نسمي هذا حدث او عندما يمر بالماوس فوق صورة ذلك حدث اخر عندما يغلق النافذة ذلك حدث ايضا و هذا جدول باهم الاحداث التي ستلزمك بالاجاكس

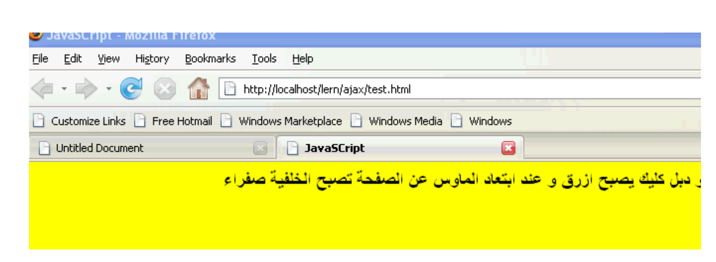
onabort	حدث ان يقوم المستخدم بالغاء شيء ما
onblur	عند ترك الحقل فارغا
onchange	عندما يغير المستخدم خيارا (بقائمة منسدلة مثلا)
onclick	ضغط واحدة يتحقق هذا الحدث
ondblclick	ضغطتين متتاليتين
ondragdrop	عندما يسحب المستخدم عنصرا و يلقيه(السحب و الالقاء)

عندما يكون هناك خطا برمجي بالجافا سكريت	onerror
عند وضع مؤشر الماوس ضمن حقل استعدادا للكتابة	onfocus
عندما يضغط المستخدم مفتاح لاسفل	onkeydown
عندما يضغط المستخدم مفتاح	onkeypress
عند تحرير مفتاح	onkeyup
يتحقق الحدث بمجرد تحميل الصفحة	onload
عندما يصبح مؤشر الماوس اسفل العنصر	onmousedown
عند تحريك الماوس	onmousemove
عندما يبتعد مؤشر الماوس بعيدا عن العنصر	onmouseout
عندما يكون مؤشر الماوس فوق العنصر مباشرة	onmouseover
عندما يحرر المستخدم زر الماوس	onmouseup
عند تصفير الخانات	onreset
عند تغيير المستخدم لحجم الصفحة او حجم عنصر	onresize
عندما يضغط المستخدم على زر سابميت	onsubmit
عند اغلاق نافذة المتصفح	onunload

عندما نطبق مثال عملي ستكتشف ان الاحداث مسلية و سهلة حيث يمكنك ان تضع اي كود جافا سكريبت بدون ان تحصره بين <script> و </script> كما تعلمنا سابقا لاحظ هذا المثال

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
</head>
<body onClick="document.bgColor='red'"
onDblClick="document.bgColor='blue' "
onMouseOut="document.bgColor='yellow'">
<h3> اضغط مرة واحدة يتغير لون الصفحة الى الاحمر و دبل كليك يصبح ازرق و عند ابتعاد الماوس عن الصفحة تصبح الخلفية صفراء </h3>
</body>
</html>
```

عندما تنفذ هذا الكود بالمتصفح سيتغير لون الصفحة الى الاحمر بمجرد الضغط عليها و الى الاصفر بابتعاد الماوس عنها



تحذير لقد استخدمنا علامتي التنصيص المفردة (' ')

صح

```
<body onClick="document.bgColor='red'">
```

ولكن اياك ان تضعها هكذا

خطأ

```
<body onClick="document.bgColor="red"">
```

اما اذا كنت تستخدمها بين عنصر سكريبت <script> و </script> فيجوز عندها استخدام العلامة المزدوجة (" ")

```
<script type="text/javascript">
document.bgColor="red";
</script>
```

الدوال(Function)

عند تنفيذك لكود دون ان تضعه داخل فنكشن سيتم تنفيذه فورا ولكن ماذا لو اردت ان ينفذ الكود فقط عندما تريد انت و ليس بمجرد تحميل الصفحة ولحل هذه المشكلة نخزن الكود داخل فنكشن وثم نستدعيه عندما نرغب بذلك.

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function hi()
{
alert(" مرحبا بك في موقعنا ");
}
</script>
</head>
<body>
</body>
</html>
```

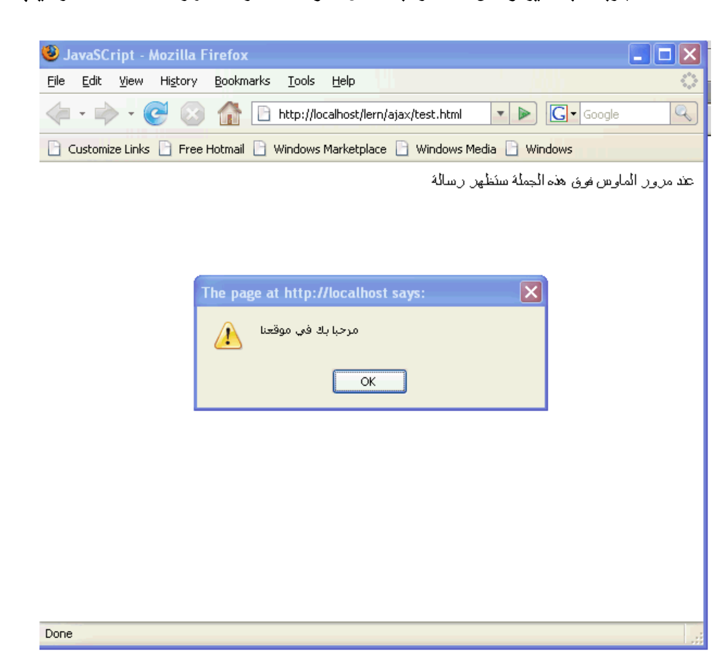
غاية في السهولة كتبنا كلمة Function
ثم اسمينا هذه الفنكشن بالاسم هاي hi

و اضفنا قوسين فاصبحت هكذا ()
و ثم احصر كودك بين الاقواس المجعدة ان صح التعبير {ضع الكود هنا}

ولكن لن يعمل الكود احزر لماذا يا شاطر؟ لاننا لم نستدعي الفنكشن بعد .
استدعاء الفنكشن سهل جدا فقط اكتب اسم الفنكشن متبوعا بقوسين لكني اريد استدعاءها فقط عندما يمر مؤشر الماوس فوق فقرة معينة

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function hi()
{
alert(" مرحبا بك في موقعنا");
}
</script>
</head>
<body>
<p onMouseOver="hi()"> عند مرور الماوس فوق هذه الجملة ستظهر رسالة </p>
</body>
</html>
```

فعندما نجربه بالفايرفوكس و تمر بالماوس فوق الفقرة تظهر لك رسالة ترحيب



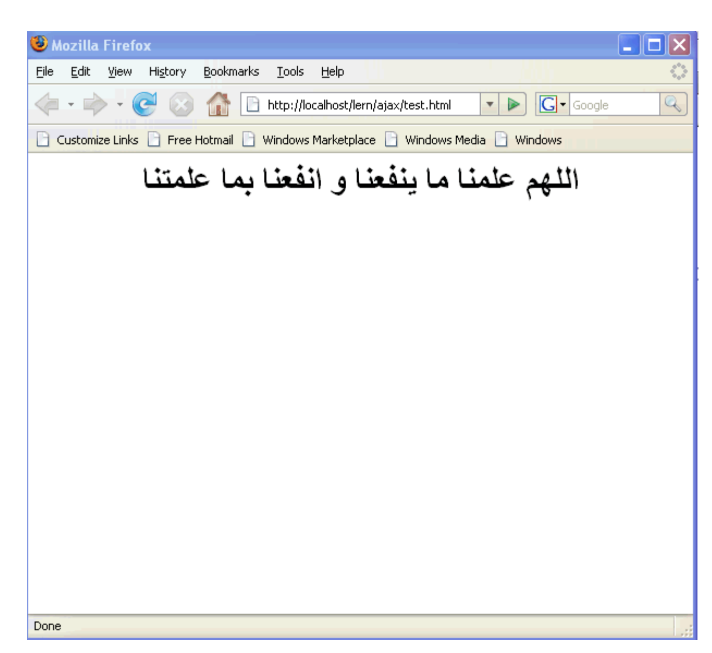
و هذا مثال اخر لاحظ انني استخدم الاحداث فعند خروج المستخدم من موقعك تظهر له رسالتي وداع الا يبدو هذا مسليا

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function good_bye()
{
alert(" الله معك");
alert("مع الف سلامة ");
}
</script>
</head>
<body onUnload="good_bye()">
</body>
</html>
```

او بامكانك ان تنفذ الكود بمجرد تحميل الصفحة و ذلك باستخدام الحدث Onload كما يلي

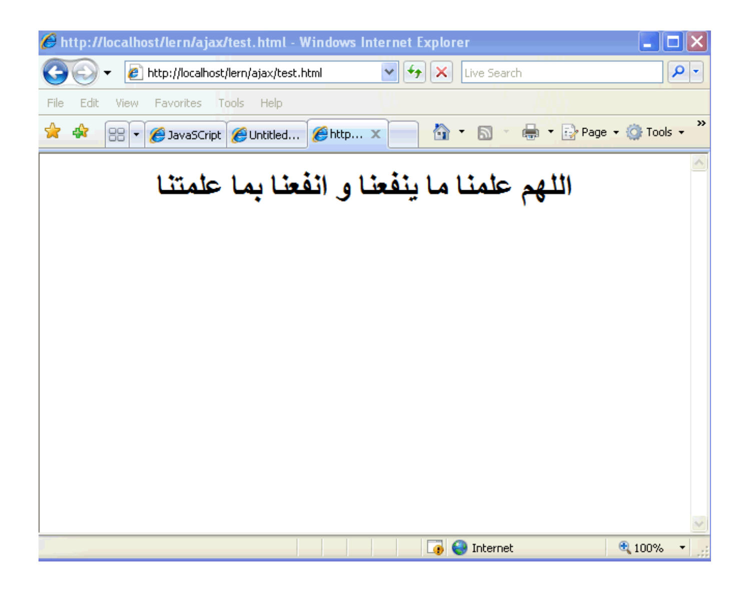
```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function text()
{
document.write(" <h1><center> اللهم علمنا ما ينفعنا و انفعنا بما علمتنا</center></h1>");
}
</script>
</head>
<body onload="text()">
</body>
</html>
```

لنرى كيف سيعرضها المتصفح :



ولكن ماذا عن محتويات الصفحة او جسم الصفحة بودي اذا كتبنا كلاما بالقسم body للاسف لن يظهر لانك عندما تستخدم الكائن document.write يقوم بمسح كل محتويات الصفحة ثم يكتب فوقها لان هذا الكائن بلغة الجافا يمثل صفحة الويب .

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<script type="text/javascript">
function text()
{
document.write(" <h1><center> اللهم علمنا ما ينفعنا و انفعنا بما علمتنا</center></h1>");
}
</script>
</head>
<body onload="text()">
<h3>للاسف هذا النص لن يظهر </h3>
<h3>بعبارة اخرى كل الصفحة لن تظهر</h3>
<h3>الذي سيظهر فقط الجملة التي كتبت بالجافا سكريبت</h3>
</body>
</html>
```

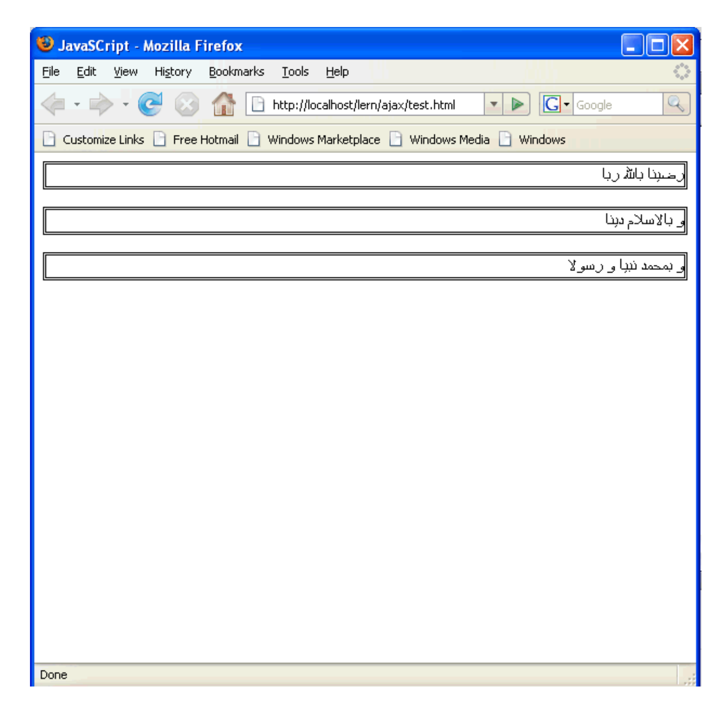


طيب لنفكر سويا بحل لهذه المشكلة . . .

اعتقد انك سمعت بالعنصر <Div>في لغة الـ HTNL استعمالها سهل جدا فهي شبيه بـ<p> ولكن ما الفرق بينها و بين الفقرة<p> الان لدينا ثلاث فقرات نريد ان نضع اطار لها باستخدام الستايل راقب الكود :

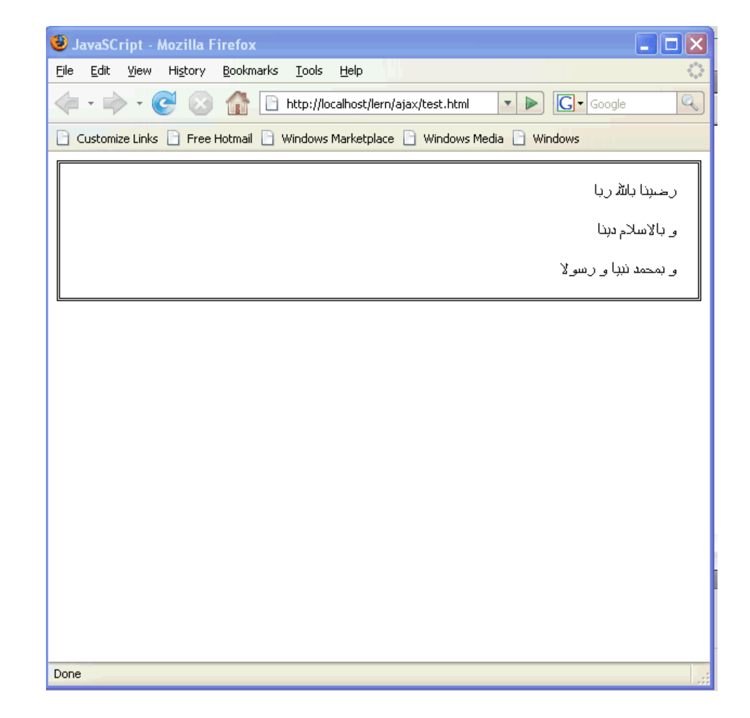
```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<style type="text/css">
.bord
{
border:double 3px;
padding-right: 20px;
}
</style>
</head>
<body>
<p class="bord">رضينا بالله ربا</p>
<p class="bord">و بالاسلام دينا </p>
<p class="bord">و محمد نبيا و رسولا</p>
</body>
</html>
```

ولكن ظهرت بشكل غير ملائم انظر الى الصورة :



و انت كنت تريد ان يكون لمجموعة الفقرات اطار واحد الحل الوحيد هو باستخدام الـ **div** الان جرب هذا الكود بنفسك بعد استخدام الـdiv

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<style type="text/css">
.bord
{
border:double 3px;
padding-right: 20px;
}
</style>
</head>
<body>
<div class="bord">
<p>رضينا بالله ربا </p>
<p> و بالاسلام دينا </p>
<p> و محمد نبيا و رسولا </p>
</div>
</body>
</html>
```



هل لاحظت الفرق ؟ و الان لنركز على خاصية مهمة تمعن في هذا الكود

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body onload="text()">
<div id="first"></div>
<div id="secondString"></div>
</body>
</html>
```

فكما ترى هذا العنصر <Div> ياخذ الخاصية ID حيث يمكنك ان تعبأ لكل<Div> خاصية IDمختلفة

فمثلا في الـ <Div>الاولى وضعنا لها الـ ID first

<div **id="first"**></div>

<div **id="secondString"**></div>

انا اركز على خاصية الـ ID لانها ستلزمنا كثيرا .

نحن تعودنا استخدام الكائن document حيث يمثل هذا الكائن صفحة الويب بأكملها و لكن ما رايك ان ننشأ كائن خاص لكل div فهذا يعطيك تحكما اكبر و مرونة بالبرنامج فمثلا الـdiv الاولى

<div **id="first"**></div> سننشا كائن خاص لها

<div **id="secondString"**></div>وكائن خاص للثانية

ولكن كيف ننشاء الكائن الخاص نحن تعودنا استخدام الخاصية **write** لكن هناك خاصية تساعدنا على تخصيص الكائن وهي getElementByIdفعندما تكتب

document.getElementById('firsr')
<div **id="first"**> فانت بذلك صنعت كائن خاص بـ

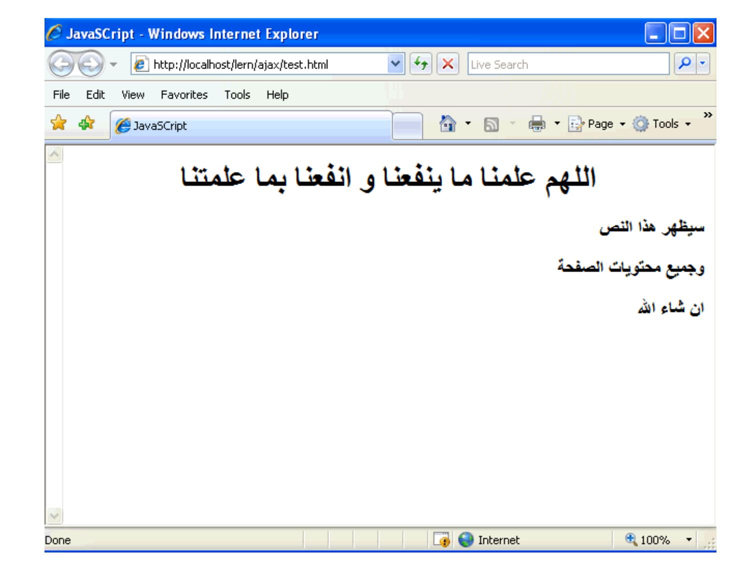
و هذا الكائن: JavaScript object corresponding to the <div> يسمونه بالانكليزي
اي كائن مقابل لتلك الـdiv

الخاصية **innerHTML**

الان بعد ان صنعت كائن خاص بـ div تريد ان ترسل نصا الى هذه الـdiv او بعبارة اخرى تريد ان تكتب نصا داخل هذه الـdiv هناك خاصية مفيدة و مساعدة جدا وهي innerHTML حيث تقوم الجافا سكربت بارسال القيمة التي تخزنها كنص يظهر في تلك الـdiv فقط اكتب :

document.getElementById("first").innerHTML

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function text()
{
document.getElementById("first").innerHTML = "<h1><center> اللهم
علمنا ما ينفعنا و انفعنا بما علمتنا</center></h1>";
}
</script>
</head>
<body onload="text()">
<div id="first"></div>
<div id="secondString"></div> <h3>سيظهر هذا النص </h3>
<h3>وجميع محتويات الصفحة</h3>
<h3>ان شاء الله</h3>
</body>
</html>
```

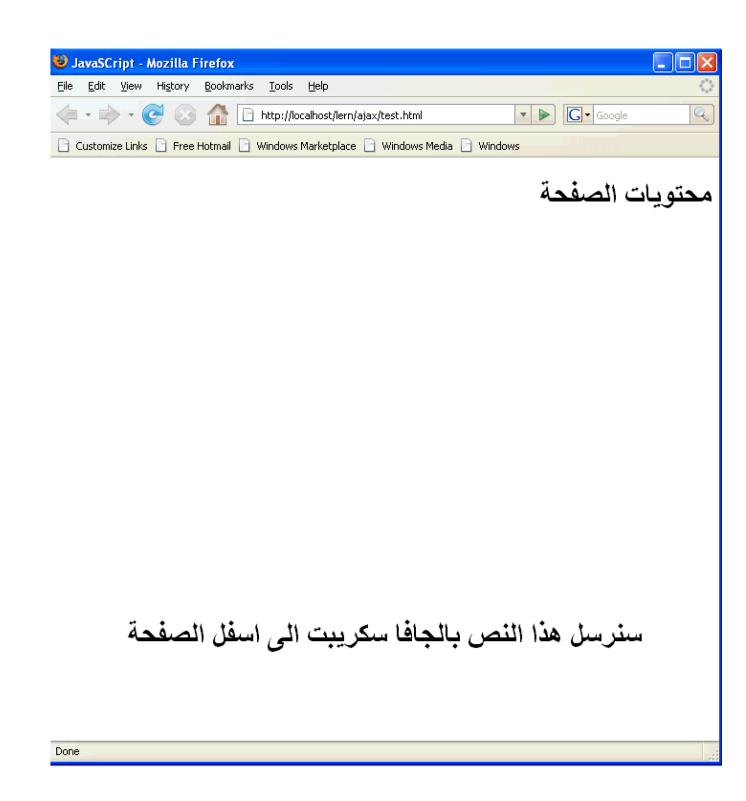


طيب لنجرب هذه المرة ان نصنع كائن اوبجكت خاص بالجملة الثانية و نرسله بالجافا سكريبت و ساضع الجملة الثانية اسفل الصفحة لكي تميزها و ساسمي الـdiv بالاسم bottom استمتع بالكود :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function text()
{
document.getElementById("bottom").innerHTML =
"<h1><center>سنرسل هذا النص بالجافا سكريبت الى اسفل الصفحة </center></h1>";
}
</script>
</head>
<body onload="text()">
<div id="first"></div>
<div id="secondString"></div>
<h1> محتويات الصفحة</h1>
<br>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>

<div id="bottom"></div>
</body>
</html>
```

كما ترى ظهرت الجملة سنرسل هذا النص بالجافا سكريبت الى اسفل الصفحة و ظهورها لم يمسح محتويات الصفحة كما حصل معنا عندما استخدمنا document.write التي تسبب مصيبة و تمسح كل شيئ



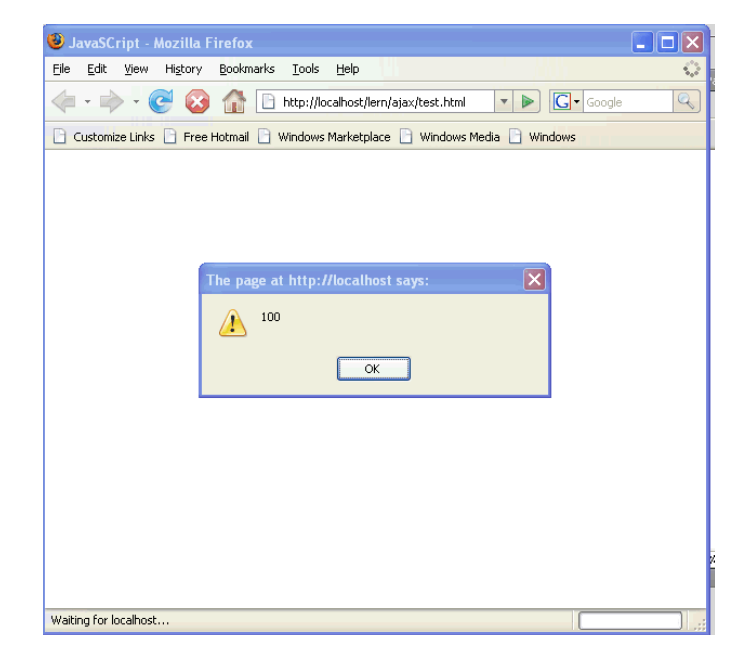
تمرير معاملات arguments

تخيل انك تريد انشاء فنكشن مهمتها ايجاد مربع العدد مثلا اذا طلبت منها مربع العشرة تعطيك مئة و في كل مرة تريد ان تعرف مربع عدد جديد هذا العدد الجديد او البيانات المتغيرة التي تدخلها الى الفنكشن نسميها argument و في المثال يتضح المقال : سنصنع فنكشن نسميها **sqr** ثم نمرر لها المعامل **num** وطلبنا منها ان تحسب مربع العدد و في كل مرة نستدعي الدالة نضع عدد مختلف

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function sqr(num)
{
alert(num*num);
}
</script>
</head>
<body onload="sqr(10)">
```

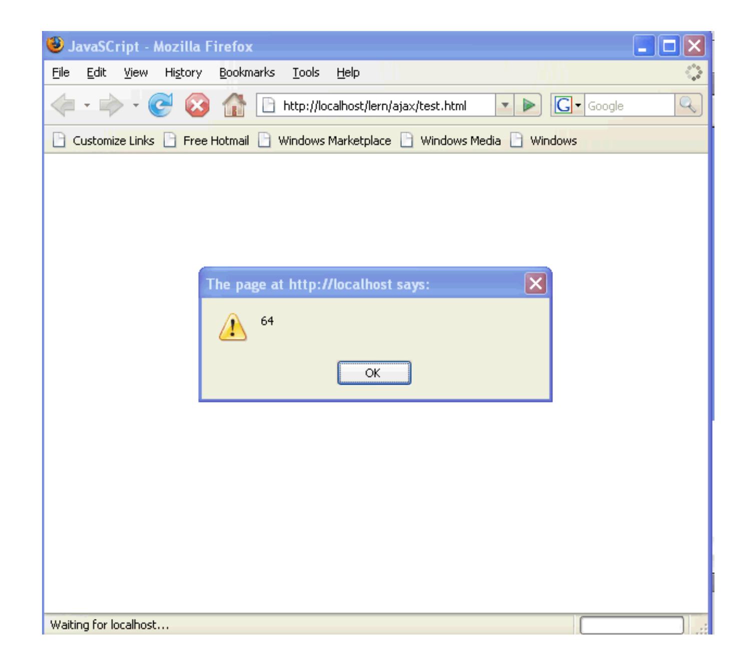


```
</body>
</html>
```



دعنا نجرب تمرير قيمة اخرى للمعامل و لتكن 8 كم سيكون الجواب يا شاطر؟

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function sqr(num)
{
alert(num*num);
}
</script>
</head>
<body onload="sqr(8)">
</body>
</html>
```

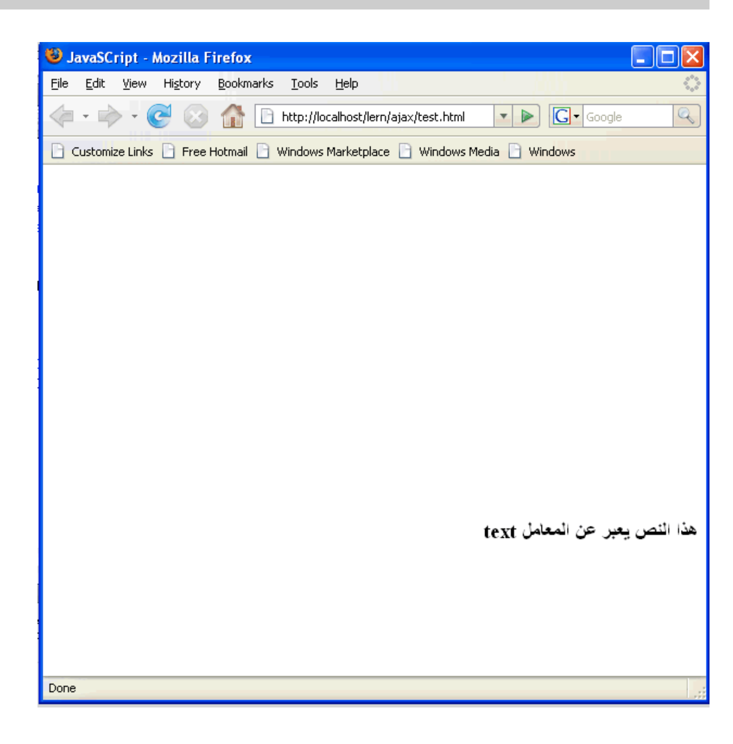


الان دعنا نطبق ما تعلمناه ولكن على العبارة:

document.getElementById("bottom").innerHTML

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function show(text)
{
document.getElementById("bottom").innerHTML=text;
}
</script>
</head>
<body onload="show('<h3> هذا النص يعبر عن المعامل text </h3>')">
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>

<div id="bottom"></div>
</body>
</html>
```

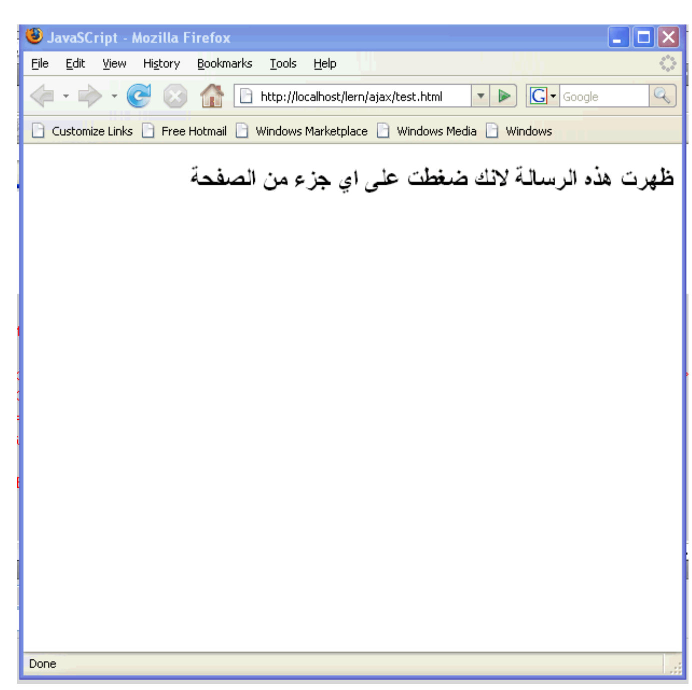


الان سامرر نص اخر و ساغير الحدث ساستخدم الحدث اون كليك فلن يظهر النص الا اذا ضغطت بالماوس على الصفحة جربه بنفسك و استمتع :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
function show(text)
{
document.getElementById("tob").innerHTML=text;
}
</script>
```

```
</head>
<body onClick="show('<h2> ظهرت هذه الرسالة لانك ضغطت على اي جزء من الصفحة</h2>')">
<div id="tob"></div>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>

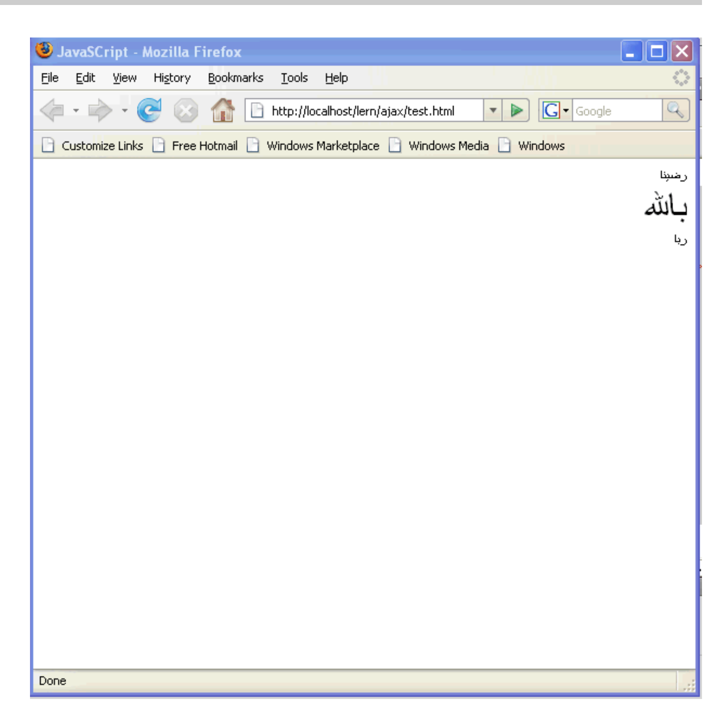
<div id="bottom"></div>
</body>
</html>
```



الفرق بين <div> و <span>

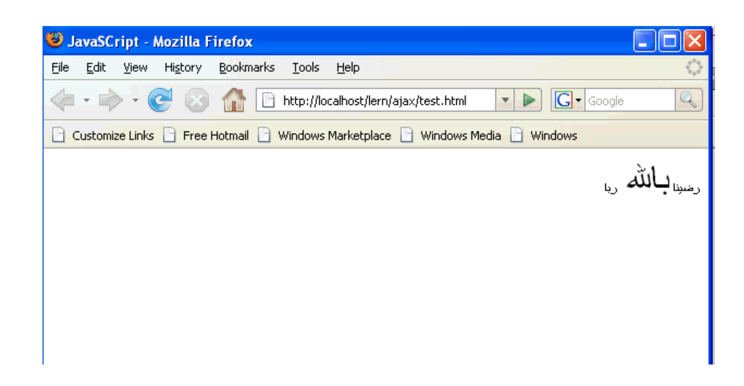
العنصر <div> يستخدم لتنسيق عدة فقرات فغالبا ما نستخدمه مع نصوص طويلة و اجزاء كبيرة من الصفحة اما عندما نتعامل مع سطر واحد عندها يجب استعمال <span> الان لديك جملة رضينا بالله ربا تريد ان تظهر لفظ الجلالة بخط اكبر لنحاول معا :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
</head>
<body>
<p>
<div style="font-size:12px">رضينا </div>
<div style="font-size:36px">بالله </div>
<div style="font-size:12px">ربا</div>
</p>
</body>
</html>
```



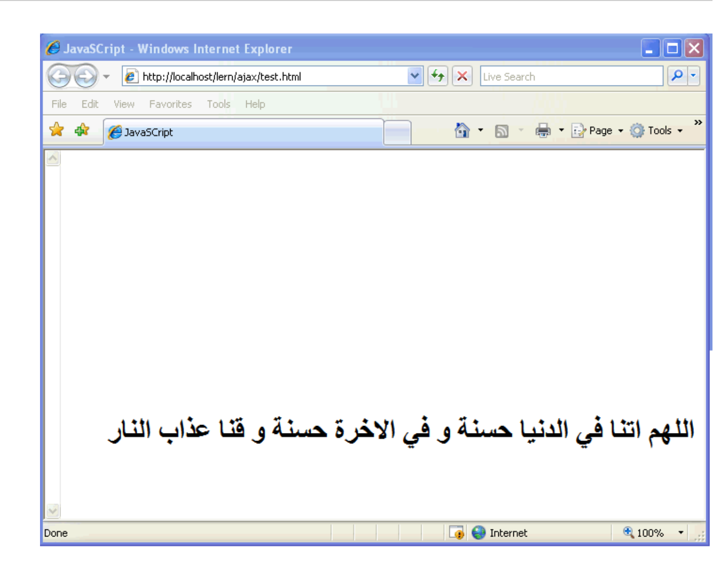
لكنك لا تريد ان تظهر كل كلمة بسطر بل تريدها جملة متلاصقة الكلمات و ياتي العنصر <span> ليحل لنا هذه المشكلة لنكتب نفس الكود لكن باستخدام <span>:

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<style type="text/css">
.bord
{
border:double 3px;
padding-right: 20px;
}
</style>
</head>
<body>
<p>
<span style="font-size:12px">رضينا </span>
<span style="font-size:36px">بالله </span>
<span style="font-size:12px">ربا</span>
</p>
</body>
</html>
```



يعني باختصار شديد استخدم <span> اذا كنت تريد ادراج نص في نفس السطر و الان سنطبق مثال متقدم يجعلك اكثر احترافية سندرج جزء من اية قرانية ثم نكمل الجزء الاخر بارساله بالجافا سكريبت :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<script type="text/javascript">
function show()
{
document.getElementById("qoran").innerHTML = "و قنا عذاب النار";
}
</script>
</head>
<body onLoad="show()">
<br> <br><br><br><br><br><br><br><br><br><br><br><br><br>
<h1>
اللهم اتنا في الدنيا حسنة و في الاخرة حسنة <span id="qoran"></span>
</h1>
</body>
</html>
```

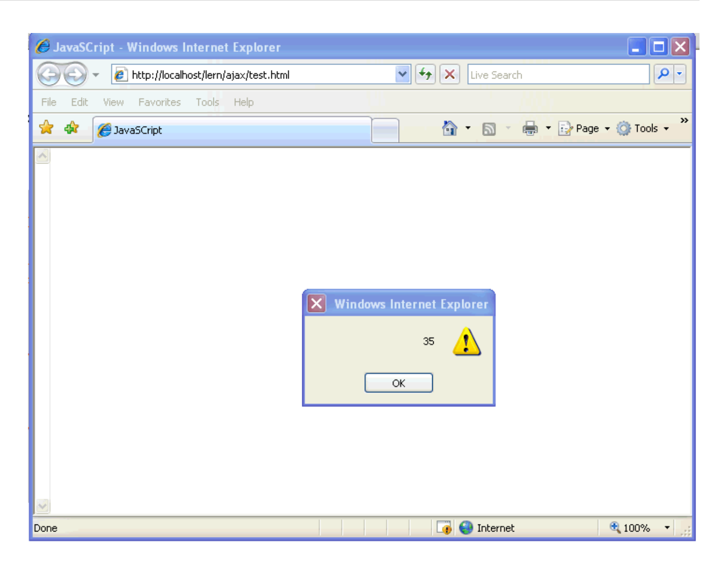


كما تلاحظ ظهر الناس الذي لرسلناه بالجافا سكريبت ملاصق تماما للنص الاصلي .

تمرير اكثر من معامل

سنصنع فنكشن تقوم بضرب عددين اذا لا بد لنا من ان نمرر لها معاملين الامر غاية في السهولة :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<script type="text/javascript">
function mult(num1,num2)
{
alert(num1*num2);
}
</script>
</head>
<body onLoad="mult(5,7)">
</body>
</html>
```

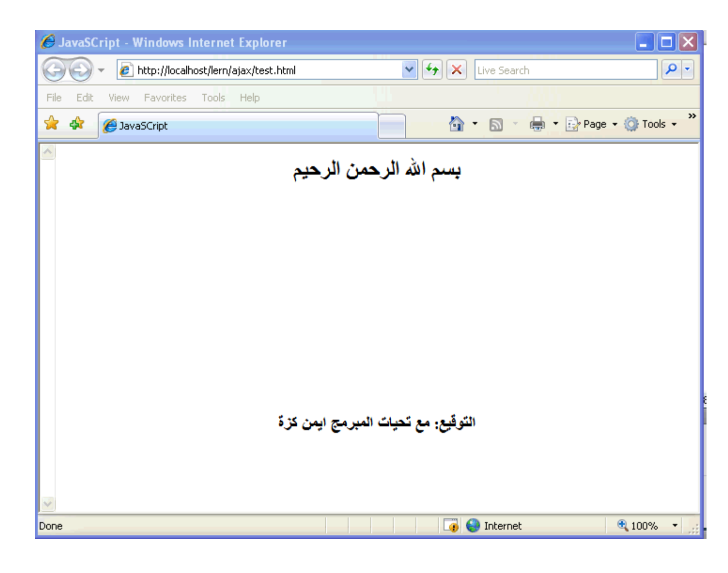


مثال اخر

الان سنصنع فنكشن جديدة و نمرر لها معاملين الاول يعبر عن النص و المعامل الثاني اين سيتوضع هذا النص لا تتكاسل من تجربة الكود بنفسك :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<script type="text/javascript">
function show(text,div_name)
{
document.getElementById(div_name).innerHTML=text;
```

```
}
</script>
</head>
<body onLoad="show('بسم الله الرحمن الرحيم','top')" onclick="show('التوقيع: مع تحيات المبرمج ايمن كزة','bottom')">
<h2> <center><div id="top"></div>
</center></h2>
<br><br><br><br><br><br><br><br><br><br><br>
<h3><center>
<div id="bottom"></div>
</center></h3>
</body>
</html>
```



طبعا التوقيع لن يظهر الا اذا ضغطت على الصفحة ولكن الفكرة الرئيسية هنا انك تتحكم بـ:

1 - النص من خلال المعامل الاول text

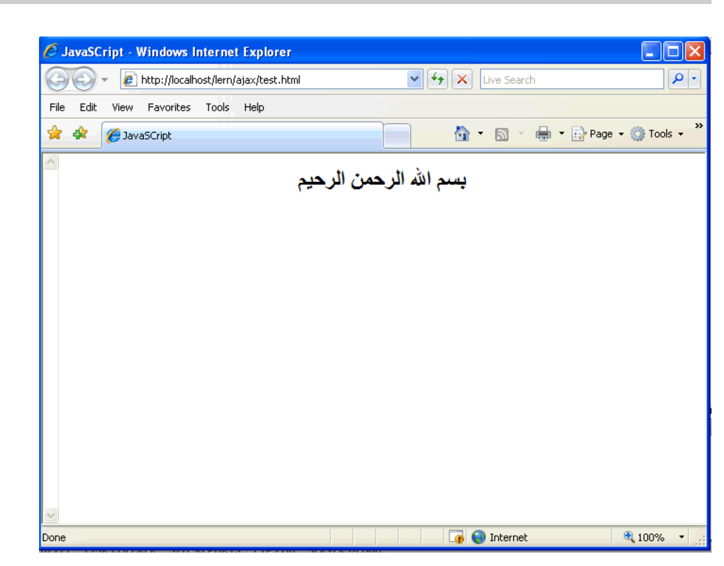
2 - مكان ظهور النص من خلال المعامل div_name

show('بسم الله الرحمن الرحيم**','top')**

المتغيرات

حيث يمكنك ان تخزن نصا او اي بيانات في متغير لكي لا تعيد كتابة ذلك النص في كل مرة فقط تكتب اسم المتغير و بالمثال يتضح المقال:

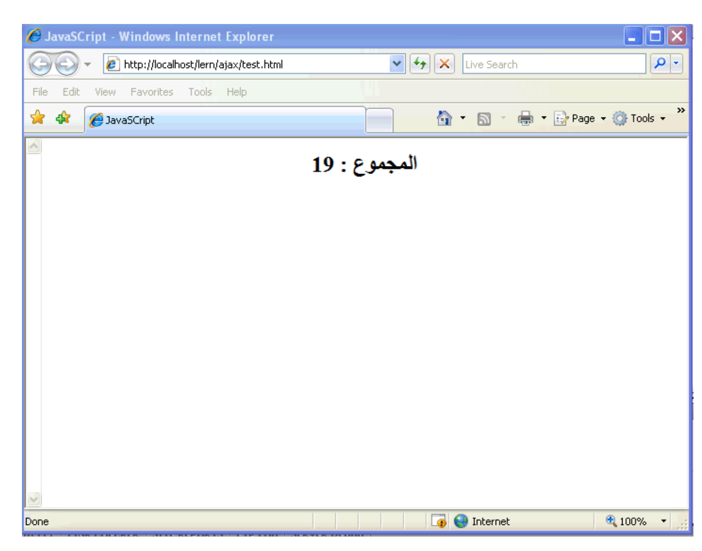
```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
var string = "بسم الله الرحمن الرحيم";
function show()
{
document.getElementById('top').innerHTML=string;
}
</script>
</head>
<body onLoad="show()">
<h2> <center><div id="top"></div>
</center></h2>
<br><br><br><br><br><br><br><br><br><br><br>
<h3><center>
<div id="bottom"></div>
</center></h3>
</body>
</html>
```



كما رايت فالنص بسم الله الرحمن الرحيم تم تخزينه في متغير اسميناه **string** الان ساعطيك مثال يقويك بالمتغيرات و تتقن استخدام المعاملات :

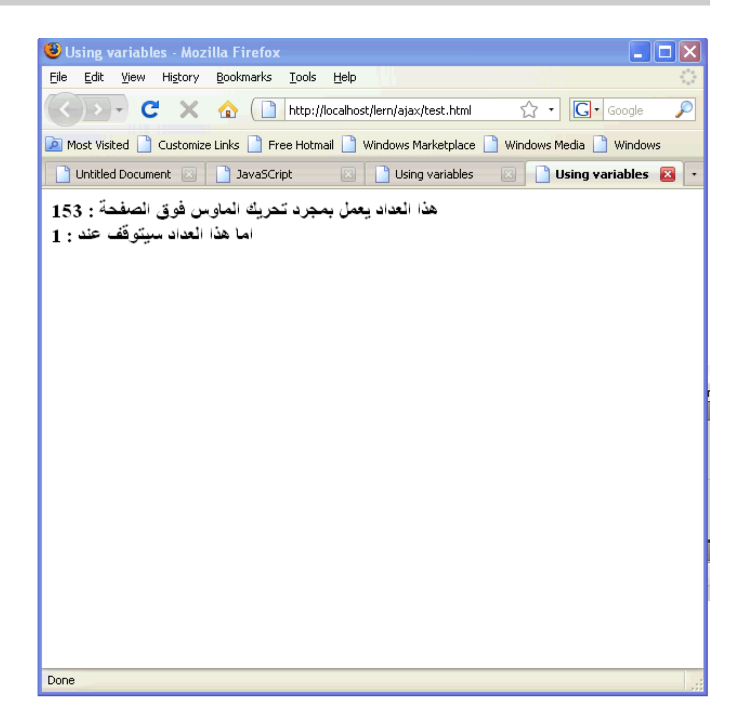
```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaSCript</title>
<script type="text/javascript">
var num1 = 5;
var num2 = 4;
var num3 = 10;
var result = 0;
function sum()
{
result = num1 + num2 + num3;
document.getElementById('top').innerHTML = result;
}
</script>
</head>
```

```
<body onLoad="sum()">
<h2><center>
المجموع : <span id="top"></span>
</center></h2>
</body>
</html>
```



لاحظ انني اعرف المتغيرات خارج الفنكشن لانك اذا عرفتها داخل الفنكشن فسيتم تصفيرها اي اعادتها الى قيمها الاصلية في كل مرة تستدعي الفنكشن :

```
<html>
<head>
<title>Using variables</title>
<script language="javascript">
var counter1 = 0;
function displayText()
{
var counter2 = 0;
counter1++;
counter2++;
document.getElementById('top').innerHTML =
" هذا العداد يعمل بمجرد تحريك الماوس فوق الصفحة : " + counter1 + "<br>" +
" اما هذا العداد سيتوقف عند : " + counter2;
}
</script>
</head>
<body onMouseOver="displayText()">
<h3>
<div id="top"></div>
</h3>
</body>
</html>
```



و سبب فشل العداد الثاني **counter2** هو اننا عرفناه داخل الفنكشن

ملاحظة

و اريدك ان تلاحظ ان المتغير النصي يجب حصره ضمن "" مثلا

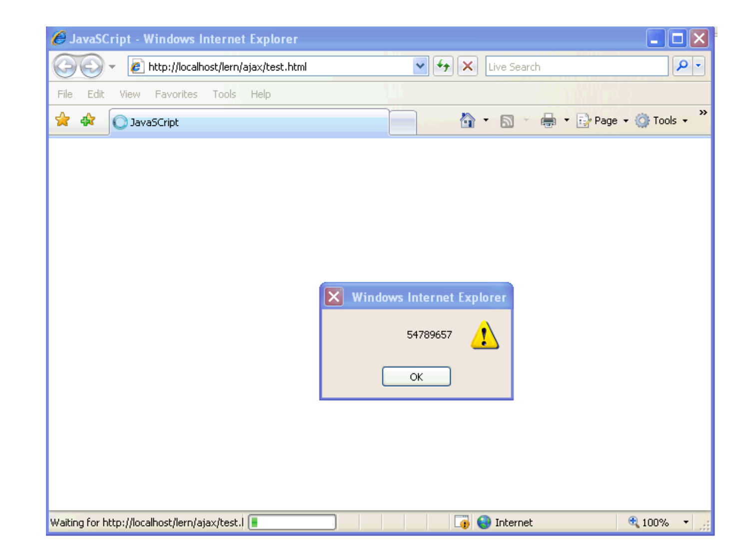
var name = "Ayman";

اما اذا كان متغير عددي فلا تضع اشارة التنصيص مثال:

var number = 15 ;

و حتى لو خزنت القيمة 15 في المتغير نومبر ثم اردت ان تغيرها الى 20 مثلا يمكنك ذلك بكل سهولة جرب الكود بنفسك :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<script type="text/javascript">
var num =15;
num = 20;
num = 0 ;
num = 54789657;
alert(num);
</script>
</head>
<body>
</body>
</html>
```



كما ترى غيرنا قيمة المتغير 4 مرات بكل سهولة

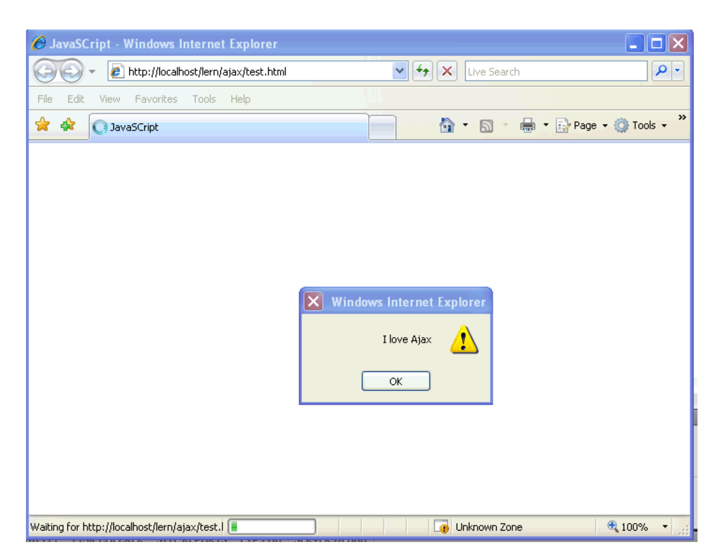
تخزين **object** في متغير

هناك الكثير من الكائنات التي مرت معنا مثل الكائن document و غيره سابقا تعلمنا تخزين نصوص او ارقام في متغير الان نريد ان نخزن كائن **object** في متغير الامر سهل جدا فقط استخدم عبارة **new** و بعدها اسم الاوبجت مثلا هناك كائن للتاريخ اسمه Date انظر كيف ساخزن هذا الكائن في متغير ساسميه **d**

```
var d = new Date();
```

الان ساخزن كائن نصي String() في متغير اسمه text_object مثلا استمتع بالكود :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript</title>
<script type="text/javascript">
var text_object = new String();
text_object = "I love Ajax";
alert(text_object);
</script>
</head>
<body>
</body>
</html>
```



الكائن XMLHttpRequest

انت كمبرمج اجاكس الكائن الذي يهمك هو الكائن **XMLHttpRequest** حيث يفيدنا هذا الكائن كثيرا عندما نسحب بيانات من السيرفر سرا (اي بدون تحديث الصفحة) و بالتالي لا يشعر المستخدم بشيء

استخدام الكائن **XMLHttpRequest**

الان لانشاء الكائن فقط اكتب مايلي :

```
var object ;

object = new XMLHttpRequest();
```

طبعا انت غير مقيد بالاسم اختر اي اسم يعجبك

```
var Ayman ;
Ayman = new XMLHttpRequest();
```

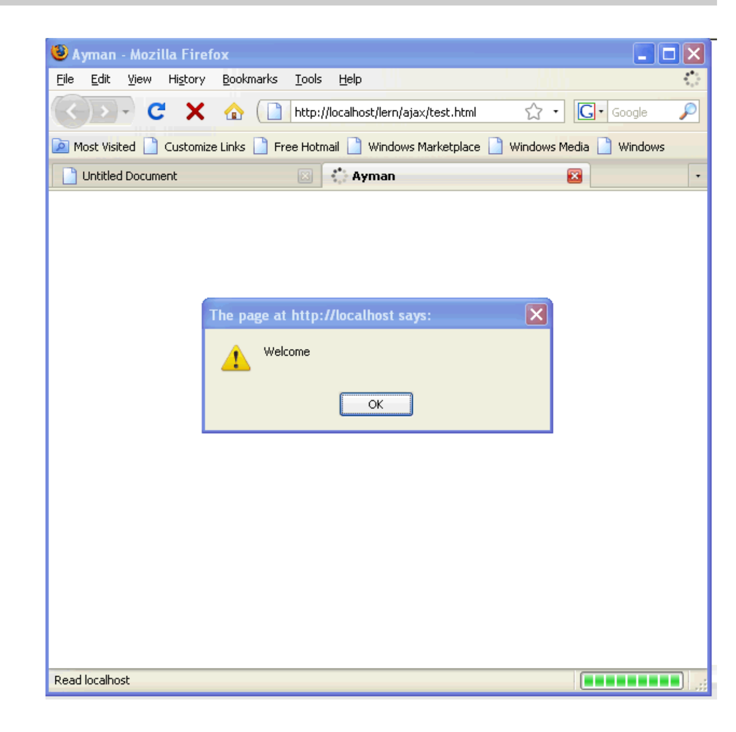
او بامكانك ان تسميه باي اسم تريد لكن درجت العادة ان يكون اسمه :

```
var XMLHttpRequestObject;
XMLHttpRequestObject = new XMLHttpRequest();
```

عبارة **if**

احيانا ترغب بان تشرط على الناس شروطا معينة فمثلا تريد انشا صفحة تطلب اسم المستخدم فاذا كان اسمك سيرحب بك و اي اسم اخر لن يتم الترحيب به انا ساضع اسمي ولكن انت لا تنسخ الكود بل ضع اسمك :

```
<html>
<head>
<title>Ayman</title>
<script language="javascript">
var name = prompt("","");
if(name == "ايمن" || name == "ayman" || name == "Ayman")
{
alert("Welcome");
}
else
{
alert("Sorry");
}
</script>
</head>
<body>
</body>
</html>
```



الكود يشرح نفسه انا وضعت شرطا اذا كان **if** الاسم ايمن او **ayman** او **Ayman** عندها اظهرت رسالة ترحيب والا **else** اظهر رسالة سوري . اريدك ان تجرب ولكن باسمك لكي تتمرن على الكود لان النسخ و اللصق لا يجدي نفعا.

معرفة المتصفح

و يتم ذلك عن طريق العبارة navigator.appName ولكني اركز الان على **if** الشرطية :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var browser = navigator.appName;
function check_borwser()
{
alert("jj");
if (browser == "Microsoft Internet Explorer")
{
document.getElementById("top").innerHTML="انت تستخدم متصفح انترنت اكسبلورر";
}
if(browser == "Netscape")
{
document.getElementById("top").innerHTML="يبدو ان متصفحك هو فايرفوكس او نت سكيب";
}
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body onload="check_borwser()">
<h3>
<div id="top"></div>
</h3>
</body>
</html>
```

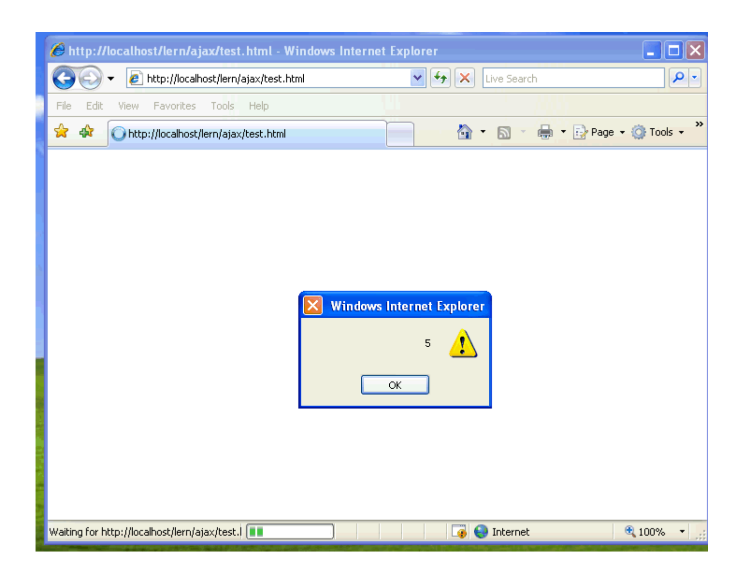
خواص النص

هناك العديد من الخواص الرائعة التي تفيدنا باقتطاع النص او معرفة عدد الحروف. . .

الخاصية **length**

تعطيك عدد حروف النص مثلا النص اجاكس ستعطيك 5 لنجرب معا:

```
<html dir="rtl">
<head>
<script type="text/javascript">
var string = "اجاكس";
alert(string.length);
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
</body>
</html>
```

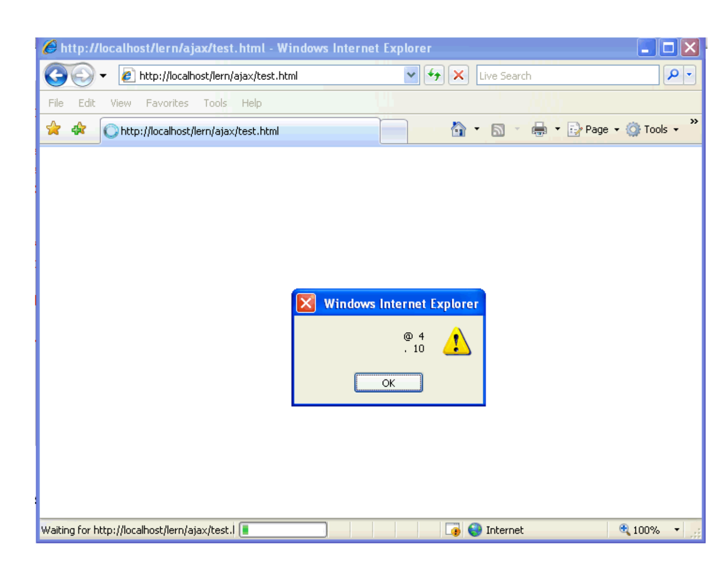


الخاصية **indexOf**

تبحث عن حرف او كلمة ثم تعطيك مكان ذلك الحرف فمثلا اذا كان اول حرف تعطيك 0 و ثاني حرف تعطي 1 وهكذا :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var mail = "name@yahoo.com";
alert(mail.indexOf("@")+" @\n"
+mail.indexOf(".")+" .");
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
</body>
</html>
```

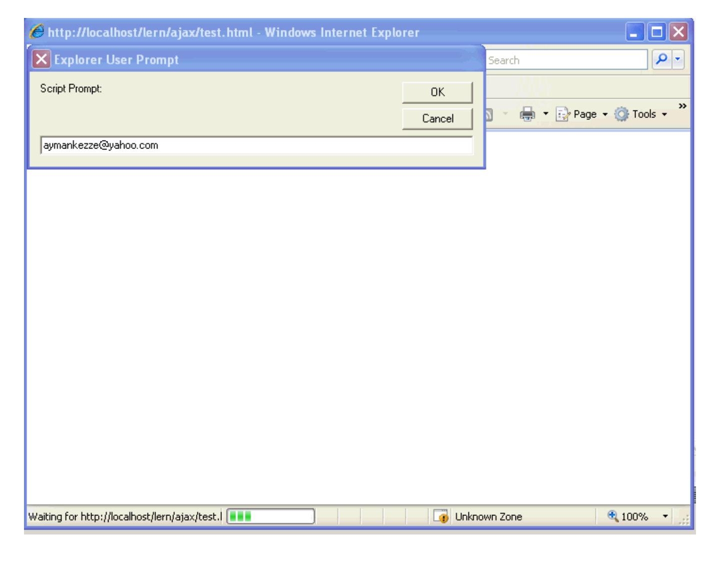
تطلب منه مكان النقطة (.) و مكان الـ(@) فكانت النتيجة :

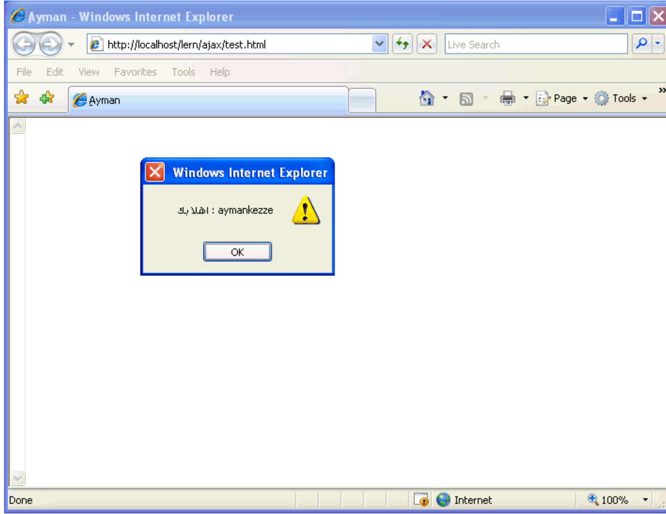


الخاصية **substr**

تطلب منك هذه الخاصية موضع الحرف البداية و موضع النهاية ثم تقتطع النص و تقدم نص جديد مثلا تريد ان تقتطع اسم صاحب الايميل لنفرض omar@hotmail.com طبعا الاسم هنا **omar** راقب هذا الكود :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var mail = prompt("","");
var at = mail.indexOf("@");
function check_mail()
{
if(at==-1) alert(" ضع ايميلك الحقيقي ");
else
{
var name=mail.substr(0,at);
alert(name+" : اهلا بك ");
}
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body onLoad="check_mail()">
</body>
</html>
```





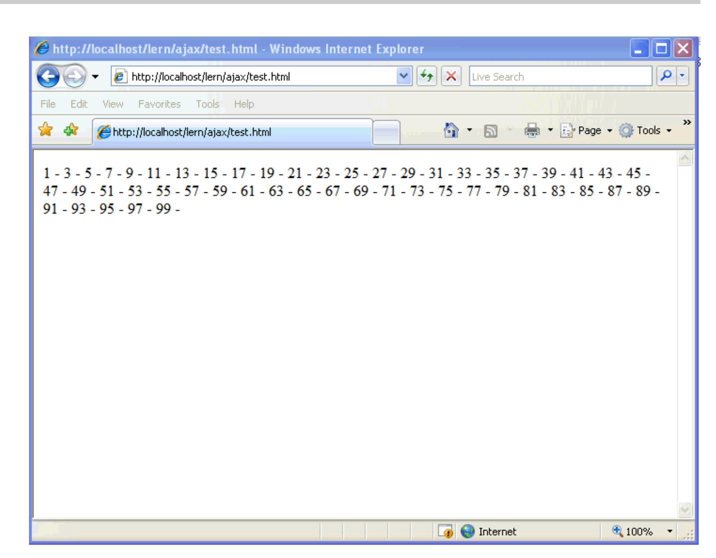
التكرار

الحلقة for

اذا ارت مثلا ان تكتب الاعداد الفدرية من 1 الى مئة سيكون ذلك صعبا ما رايك بكود يسهل الامر :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var num = 1;
var i = 0;
function loop()
{
for(i=0;num<100;i++)
{

document.write(num+" - ");
num += 2;
}
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body onLoad="loop()">
</body>
</html>
```

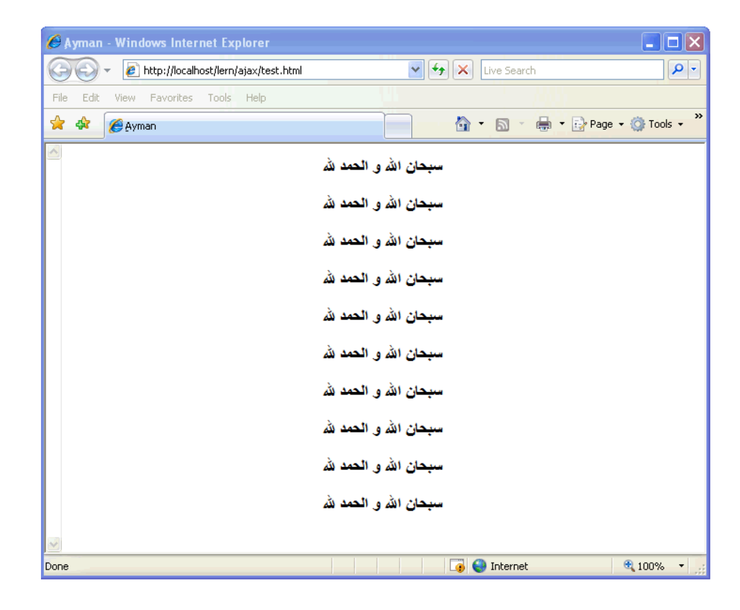


الحلقة while

و هي تنفذ ما داخل الاقواس المجعدة {} طالما الشرط صحيح و عندما يصبح الشرط خاطئ تتوقف عن التنفيذ و الشرط الذي ساضعه في مثالنا هو ان العدد **num** اصغر من عشرة تمعن بالكود جيدا :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var text = "<h3><center> سبحان الله و الحمد لله </center><br>";
var num = 0;
while(num<10)
{
document.write(text);
num++
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
</body>
```

```
</html>
```



المصفوفات

المصفوفة هي مجموعة من العناصر يمكنك ان تخزن بها كل انواع البيانات مثلا المتغيرات او القيم او الكائنات لناخذ مثال بسيط مصفوفة اسماء اصدقائك :

```
var friends = new Array("محمد","احمد","عمر","حسين");
```

طبعا المصفوفة تبدا من الصفر فعندما تطلب من الجافا سكريبت ان تطبع القيمة صفر من مصفوفة الاصدقاء ستعطيك محمد

alert(friends[0]); سيكون الناتج محمد

alert(friends[1]); سينتج احمد

alert(friends[2]); سينتج عمر

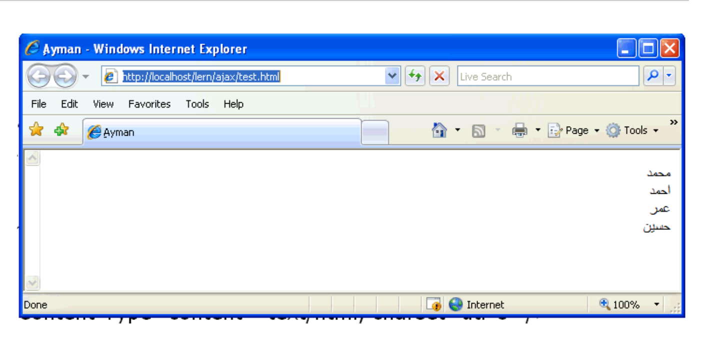
اما اذا اردت عدد عناصر المصفوفة استخدم الخاصية **length** فهنا لديك اربع عناصر

alert(friends.length); **4** ستعطيك

طباعة عناصر المصفوفة

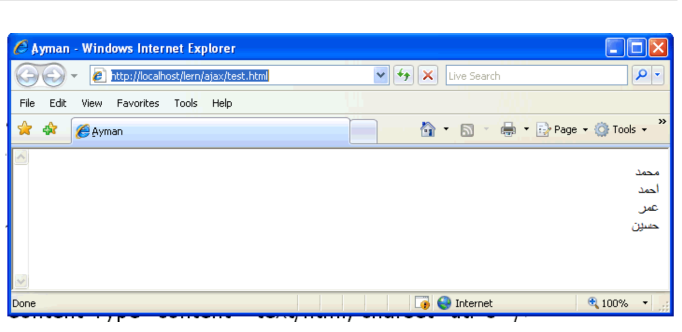
ويتم ذلك باستخدام الحلقة فور لنشاهد الكود :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var friends = new Array("محمد","احمد","عمر","حسين");
for(i=0;i<friends.length;i++)
{
document.write(friends[i]+"<br>");
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
</body>
</html>
```



كما يمكنك طباعة عناصر المصفوفة باستخدام حلقة **while**

```
<html dir="rtl">
<head>
<script type="text/javascript">
var friends = new Array("محمد","احمد","عمر","حسين");
var i = 0 ;
while(i<friends.length)
{
document.write(friends[i]+"<br>");
i++;
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
</body>
</html>
```

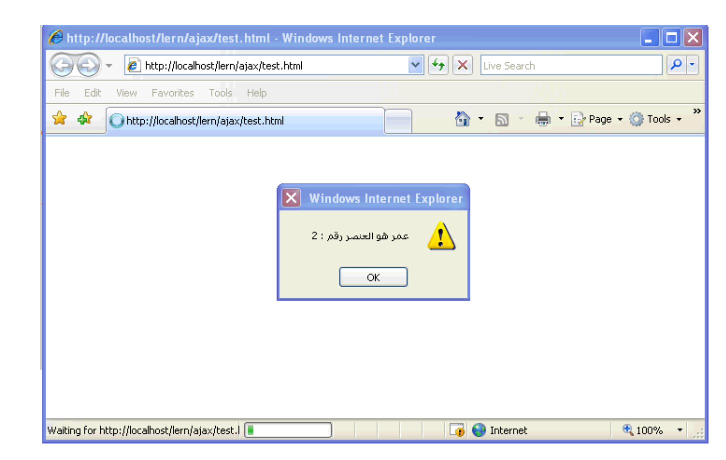


كما يمكنك انشا المصفوفة بطريقة ثانية :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var friends = new Array(4);
friends[0] = "محمد";
friends[1] = "احمد";
friends[2] = "عمر";
friends[3] = "حسين";
var i = 0 ;
while(i<friends.length)
{
document.write(friends[i]+"<br>");
i++;
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
</body>
</html>
```

احيانا يلزمنا ان نعرف رقم العنصر في المصفوفة طبعا كما تعلم العنصر الاول رقمه 0 و الثاني 1 و الثالث رقمه 2 الان لنبحث عن رقم العنصر عمر في المصفوفة

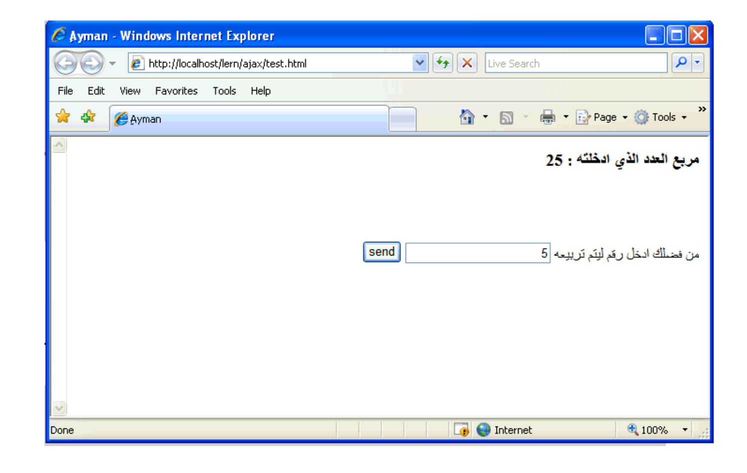
```
<html dir="rtl">
<head>
<script type="text/javascript">
var friends = new Array(4);
friends[0] = "محمد";
friends[1] = "احمد";
friends[2] = "عمر";
friends[3] = "حسين";
var i = 0 ;
while(i<friends.length && friends[i] != "عمر")
{
i++;
}
if(i<friends.length) alert("عمر هو العنصر رقم : "+i);
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
</body>
</html>
```



التعامل مع النماذج

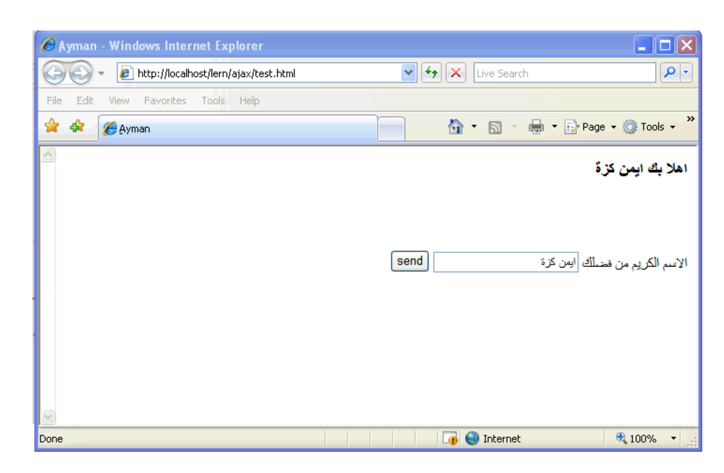
انا لا احب الشرح كثيرا لانه غير مفيد بالبرمجة الذ شيء هو التطبيق العملي اول تطبيق عملي عن الفورم ساقدمه لك عزيزي المتدرب هو برنامج يحسب مربع العدد تفضل:

```
<html dir="rtl">
<head>
<script type="text/javascript">
function math()
{
var num = document.getElementById("num_id").value;
var sqr = num*num;
document.getElementById("top").innerHTML = " مربع العدد الذي ادخلته :
"+sqr;
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
<h3>
<div id="top"></div>
</h3><br><br>
<form name="form1">
من فضلك ادخل رقم ليتم تربيعه
<input type="text" id="num_id">
<input type="button" onClick="math()" value="send">
</form>
</body>
</html>
```



و الان سنطلب من المستخدم كتابة اسمه لتظهر جملة ترحب به :

```
<html dir="rtl">
<head>
<script type="text/javascript">
function show()
{
var string = document.getElementById("text_id").value;
document.getElementById("top").innerHTML = " اهلا بك "+string;
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
<h3>
<div id="top"></div>
</h3><br><br><br>
<form name="form1">
الاسم الكريم من فضلك
<input type="text" name="user_name" id="text_id"/>
<input type="button" onClick="show()" value="send">
</form>
</body>
</html>
```



ولكن كيف تمكننا من جلب الاسم؟ انتبه الى الحقل النصي الذي يحمل الايدي :

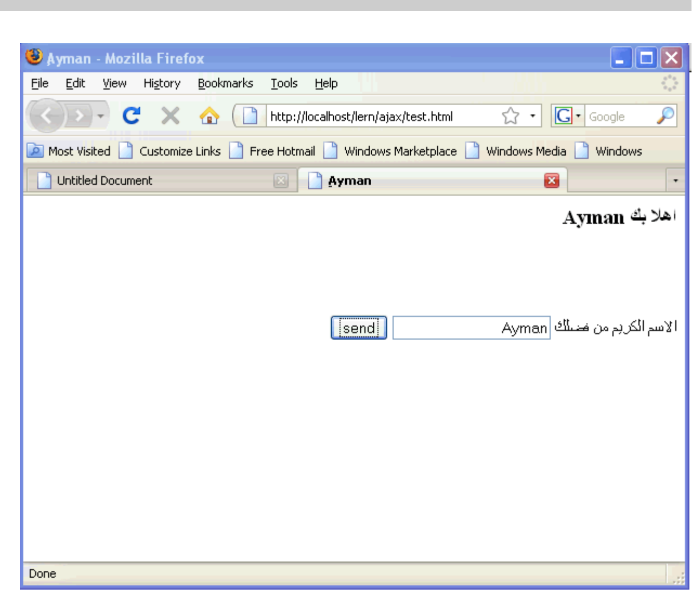
```
<input type="text" name="user_name" id="text_id"/>
```

و طالما عرفنا الايدي خزننا هذا النص في متغير اسميناه سترينغ:

```
var string =
document.getElementById("text_id").value;
```

كما يمكننا جلب المعلومات التي يكتبها المستخدم بطريقة ثانية عن طريق اسم الفورم و اسم الحقل شاهد الكود :

```
<html dir="rtl">
<head>
<script type="text/javascript">
function show()
{
var string = form1.user_name.value;
document.getElementById("top").innerHTML = " اهلا بك "+string;
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman</title>
</head>
<body>
<h3>
<div id="top"></div>
</h3><br><br><br>
<form name="form1">
الاسم الكريم من فضلك
<input type="text" name="user_name"/>
<input type="button" onClick="show()" value="send">
</form>
</body>
</html>
```



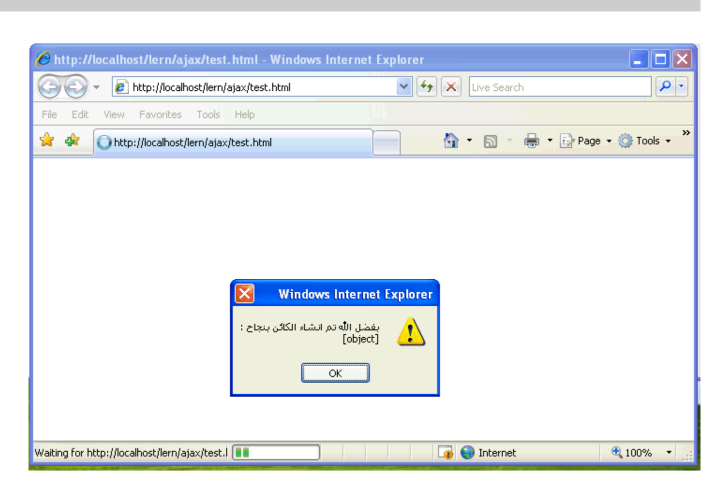
الفصل الثاني اجاكس

و اخيرا وصلنا الى مبتغانا بفضل الله , تمكنك تقنية اجاكس من جلب المعلومات سواء كانت نصية او اكس ام ال بدون تحديث الصفحة و بدون ان يلاحظ المستخدم ان الصفحة اتصلت بالسيرفر و هذا ما يجعل برمجياتك متميزة عن التطبيقات الاخرى فالمبرمج الذي يستخدم تقنية الاجاكس في سكربتاته يعتبر مبرمج محترف

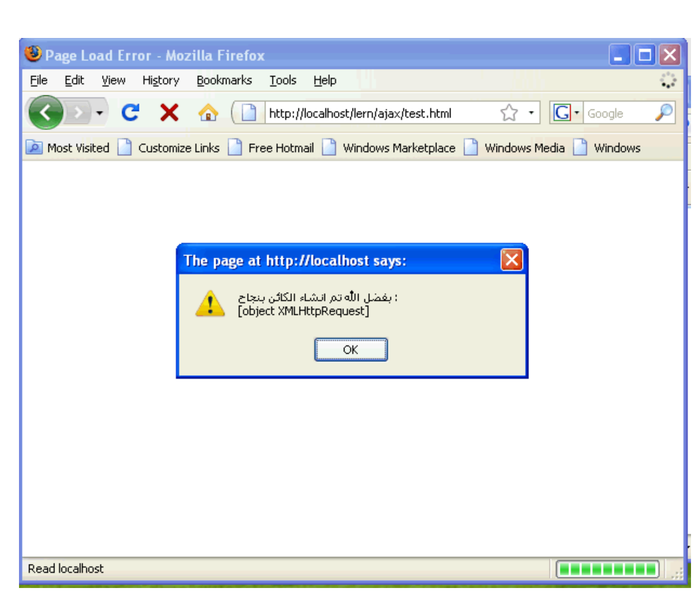
الكائن XMLHttpRequest

حسنا الان كلنا عرفنا الاجاكس تقنية رائعة ولكن كيف تقوم الاجاكس بسحب بيانات من السيرفر و عرضها بالصفحة بدون اعادة تحميل الصفحة ؟ الحقيقة ان ذلك يتم باستخدام الكائن XMLHttpRequestObject و هذا الكائن لديه العديد من الخواص المفيدة التي سناتي على ذكرها لاحقا و الان لننشئ هذا الكائن و خزنه في متغير و ليكن اسمه ajax

```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false;
if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
alert("بفضل الله تم انشاء الكائن بنجاح :\n" +ajax);
}
else
{
alert("متصفحك لا يدعم هذا الكائن");
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>بدأنا بالاجاكس </title>
</head>
<body>
</body>
</html>
```



و الان لنجربه على الفايرفوكس



اذا كما رايت انشاء الكائن سهل جدا فقط اكتب العبارة

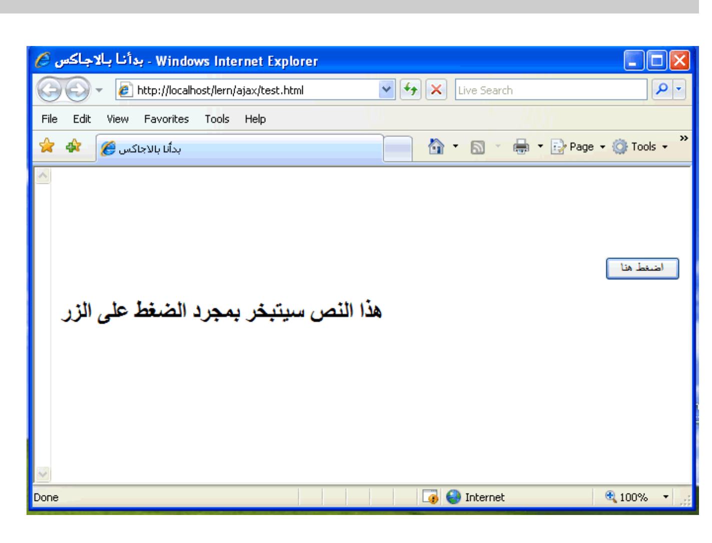
```
ajax = new XMLHttpRequest();
```

اما هذا :if(window.XMLHttpRequest) للتحقق من ان المتصفح يدعم انشاء الكائن السطر الذي يحوي الشرط

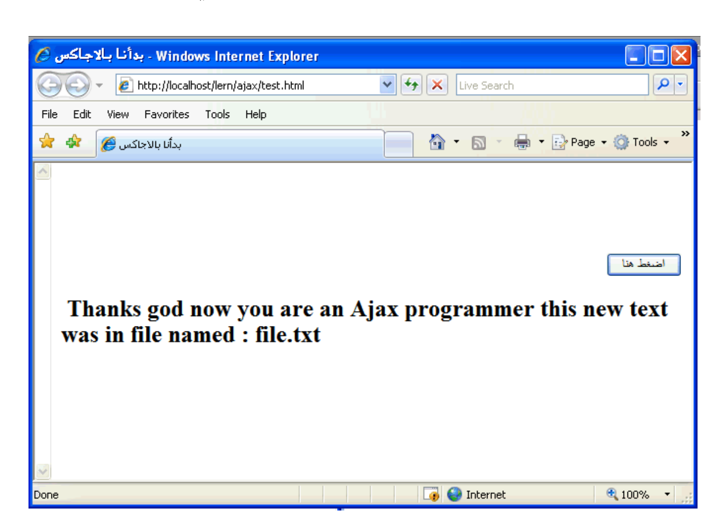
الان اصبح الكائن جاهز و مخزن في متغير اسمه ajax الان ساضع كود كامل يسحب البيانات من السيرفر بطريقة سهلة و باستخدام تقنية الاجاكس , لا تنزعج من طول الكود لانني ساحلله تحت المجهر و سأشرحه لك سطرا سطرا تفضل اقرأ الكود اولا ثم الشرح:

```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false
ajax = new XMLHttpRequest();
//____________________________________________________
function grab_data(data_sourse,div_id)
{
if(ajax)
{
var ob = document.getElementById(div_id);
ajax.open("get",data_sourse);
ajax.onreadystatechange = function()
// start new function body
{

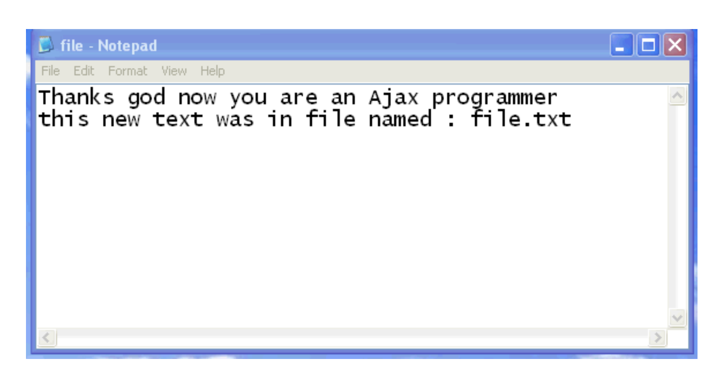
if(ajax.readyState == 4 && ajax.status == 200)
{
ob.innerHTML = ajax.responseText;
}
}
// End new function body
ajax.send(null);
}
}
//______________________________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>بدأنا بالاجاكس </title>
</head>
<body>
<br><br><br><br>
<form>
<input type="button" value=" اضغط هنا "
onClick="grab_data('file.txt','left_div')"/>
</form>
<h2>
<div align="left" id="left_div"> هذا النص سيتبخر بمجرد الضغط على الزر </div>
</h2>
</body>
</html>
```



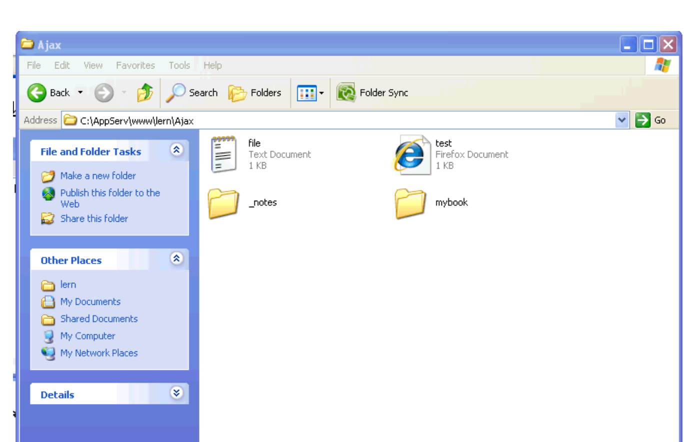
و بمجرد الضغط على الزر سيظهر النص الجديد المخزن في الملف :



كما ترى تم استبدال النص القديم بنص جديد مكتوب بملف اسمه file.txt



ويجب ان تضع الملف النصي file.txt و ملف الذي يحوي كود الاجاكس بنفس المجلد :



ربما تقول ما الجديد في ذلك فقط عرضنا نص؟ لا طبعا انت الان سحبت معلومات من السيرفر بتقنية الاجاكس بدون تحديث الصفحة الان لنحلل الكود بسم الله و لنبدأ بالبودي قبل الهيد لانه الاسهل, اولا نرى النص الاصلي مخزن في **div** طبعا هذا النص سيطير و يتم استبداله لكن المهم ان هذه الـ **div** تحمل الاسم **left_div** :

```
<body>
<br><br><br>
<form>
<input type="button" value=" اضغط هنا "
onClick="grab_data('file.txt','left_div')"/>
</form>
<h2>
<div align="left" id="left_div"> هذا النص سيتبخر بمجرد الضغط على الزر </div>
</h2>
</body>
```

و هناك ايضا زر مهمته ان يستبدل النص القديم باخر جديد حيث استخدمنا الحدث اون كليك و الفنكشن grab_data اي ان الفنكشن سيتم استدعائها اذا ضغط المستخدم على الزر و مررنا قيمتين لهذه الفنكشن اول قيمة هي اسم الملف **file.txt**و الثانية اسم الـ **div** و هي **left_div**:

```
<body>
<br><br><br>
<form>
<input type="button" value=" اضغط هنا "
onClick="grab_data('file.txt','left_div')"/>
</form>
<h2>
<div align="left" id="left_div"> هذا النص سيتبخر بمجرد الضغط على الزر </div>
</h2>
</body>
```

و اريد ان احذرك من خطأ بسيط لكنه خطير و يقع فيه اقوى المبرمجين بسبب السرعة فاياك ان تكتب السطر السابق بهذه الطريقة

```
onclick="grab_data("file.txt","left")" /> خطا
```

```
onClick="grab_data('file.txt','left_div')"/> صح
```

و الان لننتقل القسم هيد قبل ان نبدا الفنكشن نريد انشاء الكائن بكل سهولة كما تعلمنا :

```
var ajax = false
ajax = new XMLHttpRequest();
```

اما لماذا خزننا القيمة البولونية false بهذا الكائن لكي نتحقق فيما بعد هل تم انشاؤه بنجاح .

الان و بعد ان اصبح الكائن جاهز كيف نقوم بجلب البيانات من السيرفر لا بد من صناعة فنكشن ما رايك ان نسميها grab_data و سنمرر لها معاملين الاول اسم الملف data_sourse اما المعامل الثاني فهو div_id :

```
function grab_data(data_sourse,div_id)
```

اول ما سنكتبه في هذه الفنكشن هو التحقق من ان المتغير ajax هل يحمل فعلا الكائنXMLHttpRequest او بمعنى اخر هل تم انشاء الكائن بنجاح فلا بد من عبارة **if** الشرطية :

```
function grab_data(data_sourse,div_id)
{
if(ajax)
{
-
-
-
}
```

السطر التالي سهل جدا و مر معنا بالفصل السابق عشرات المرات :

```
var ob = document.getElementById(div_id);
```

الان وصلنا الحزء المهم لدينا خاصية مفيدة جدا تستخدم مع هذا الكائن و الخاصية **open** حيث نمرر لهذا الخاصية معاملين او بارامترين البراميتر الاول يمثل الطريقة التي ستسحي بها البيانات هل هي GET ام POST اما البراميتر الثاني نضع فيه اسم او مسار الملف data_sourse

لكي لا تحتار ساخبرك بالسر عندما تستقبل بيانات من السيرفر استخدم GET اما اذا اردت ان ترسل بيانات استخدم POST :

```
ajax.open("get",data_sourse);
```

و هذا الكائن لديه خاصية تراقب حالة الكائن XMLHttpRequest تدعى:
onreadystatechange عندما تسند هذه الخاصية لفنكشن بالجافا سكريبت فسيتم استدعاء تلك الفنكشن في كل مرة تتغير حالة الكائن فمثلا عند ارسال بيانات تتغير حالة الكائن و كذلك عندما تسحب بيانات تتغير حالة هذا الكائن و بالتالي يتم استدعاء الفنكشن . نحن تعودنا ان نصنع الفنكشن بهذه الطريقة

```
function fun_name()
{
-
-
}
```

اي نكتب كلمة function ثم الاسم الذي نختاره نحن اما الان فسنصنع فنكشن بدون اسم يطلق عليها المبرمجون اسم الفنكشن المجهولة لانها بدون اسم:

```
function grab_data(data_sourse,div_id)
{
if(ajax)
```

```
{
var ob = document.getElementById(div_id);
ajax.open("get",data_sourse);
ajax.onreadystatechange = function()
// start new function body
{
-
-
}
// End new function body
-
-
}
```

الان و بعد انا صنعنا الفنكشن المجهولة و اسندنا اليها خاصية onreadystatechange سنكتب جسم الفنكشن الجديدة سيحتوي جسم الفنكشن الجديدة على شرطين و بعد تحقق هذين الشرطين سيتم استقبال البيانات :

```
function grab_data(data_sourse,div_id)
{
if(ajax)
{
var ob = document.getElementById(div_id);
ajax.open("get",data_sourse);
ajax.onreadystatechange = function()
// start new function body
{
if(ajax.readyState == 4 && ajax.status == 200)
{
ob.innerHTML = ajax.responseText;
}
}
// End new function body
ajax.send(null);
}
}
```

الشرط الاول **ajax.readyState == 4** دعني اشرح لك الخاصية**readyState** هذه الخاصية تخبرك عن مدى اكتمال تحميل البيانات فربما انت تحمل صور ضخمة و تاخذ هذه الخاصية القيم التالية:

0 uninitialized
1 loading
2 loaded
3 interactive
4 complete

الذي يهمنا هو **4 complete** اي عندما تكون قيمة readyState تساوي 4 فهذا يعني ان التحميل قد تم.

اما الشرط الثاني :**ajax.status == 200**

الخاصية **status** ايضا احد خواص الكائن XMLHttpRequest و تاخذ القيم التالبة :

200 OK
201 Created
204 No Content
205 Reset Content
206 Partial Content
400 Bad Request
401 Unauthorized
403 Forbidden
404 Not Found
405 Method Not Allowed
406 Not Acceptable
407 Proxy Authentication Required
408 Request Timeout
411 Length Required
413 Requested Entity Too Large
414 Requested URL Too Long
415 Unsupported Media Type
500 Internal Server Error
501Not Implemented
502 Bad Gateway
503 Service Unavailable
504 Gateway Timeout
505 HTTP Version Not Supported

فعندما تحمل الخاصية status القيمة200 فهذا يعني ان كل شي تمام التمام كما يقولون باللهجة المصرية

الان اصبحت البيانات جاهزة تماما لم يبقى الا ان نستقبلها و ثم نعرضها في الصفحة

استقبال البيانات

اذا كنت تستقبل بيانات نصية استخدم الخاصية **responseText**
اما اذا كنت تستقبل بيانات XML استخدم الخاصية **responseXML**

و بما انك تستقبل بيانات نصية في مثالنا هنا طبعا ستستخدم responseText كالتالي:

```
function grab_data(data_sourse,div_id)
{
if(ajax)
{
var ob = document.getElementById(div_id);
ajax.open("get",data_sourse);
ajax.onreadystatechange = function()
// start new function body
{
if(ajax.readyState == 4 && ajax.status == 200)
```

```
{
ob.innerHTML = ajax.responseText;
}
}
// End new function body
ajax.send(null);
}
}
```

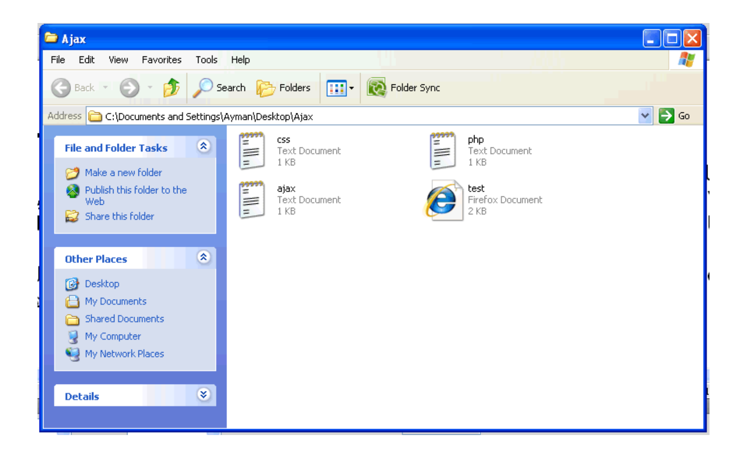
الان وصل ساعي البريد و يطرق بابك لتستلم الطرد البريدي لم يبق الا ان تفتح الباب و تستلم تلك البيانات . الخطوة الاخيرة و التي شبهتها لك بفتح الباب هي الخاصية **send** فاذا كنت قد استخدمت الطريقة جيت مرر لها القيمة **null** اما عندما تريد ارسال بيانات بالطريقة بوست عندها مرر لها قيمة نصية انظر كيف تستخدم هذه الخاصية:

```
function grab_data(data_sourse,div_id)
{
if(ajax)
{
var ob = document.getElementById(div_id);
ajax.open("get",data_sourse);
ajax.onreadystatechange = function()
// start new function body
{
if(ajax.readyState == 4 && ajax.status == 200)
{
ob.innerHTML = ajax.responseText;
}
}
// End new function body
ajax.send(null);
}
}
```

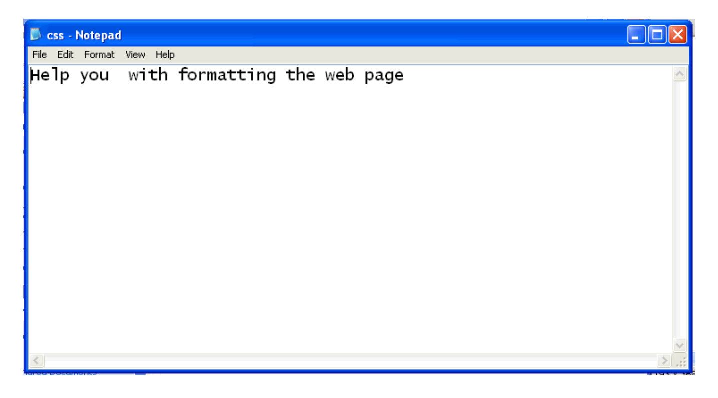
مبرووووووووووووك الان تستطيع ان تقول باعلى صوتك انا مبرمج اجاكس طالما انهيت هذا الكود لن يصعب عليك شيء ان شاء الله

تطبيق اخر

كثرة الامثلة و التطبيقات عندما تجرب و تكتب الكود بيدك بدون نسخ و لصق هو ما يجعلك مبرمج محترف الان سنكتب معا سكربت اعتمادا عل نفس الدالة و لكن الان لدي ثلاثة ملفات نصية و كل ملف مخصص ليتم عرضه كشرح موجز للاسم الذي تمر فوقه الماوس مثلا عند مرور الماوس فوق كلمة تقنيةالاجاكس فسوف يظهر شرح لهذه التقنية بالانكليزي Help you to fetch data: from the server without reloading the page و هكذا الان نحتاج الى ثلاثة ملفات نصية ضعها بنفس المجلد :



و في كل ملف اكتب شرح مختصر :



الان لم يبق علينا الا الكود تفضل :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false;
ajax = new XMLHttpRequest();
//________________
function grab_data(data_sourse,div_id)
{
if(ajax)
{
var ob = document.getElementById(div_id);
ajax.open("get",data_sourse);
ajax.onreadystatechange = function ()
// start new function
{
if(ajax.readyState == 4 && ajax.status == 200)
{
ob.innerHTML = ajax.responseText;
}
}
// End new function
ajax.send(null);
}
}
//________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<table width="500" border="0" align="center" cellpadding="3"
cellspacing="3">
<tr>
```

```
<td width="33%" align="center"
onmouseover="grab_data('ajax.txt','details')"><h1>تقنية الاجاكس
</h1></td>
<td width="33%" align="center"
onmouseover="grab_data('css.txt','details')"><h1>CSS </h1></td>
<td width="34%" align="center"
onmouseover="grab_data('php.txt','details')"><h1>php</h1></td>
</tr>
</table>
<p> </p><font size="4" color="#993300">
<div id="details" dir="ltr"></div>
</font>
</body>
</html>
```

ليس هناك جديد سوى اننا انشانا جدول ثم وضعنا الحدث **onmouseover** لكي يتم شرح الكلمة التي تمر الماوس فوقها :

onmouseover="grab_data('ajax.txt','details')"

و الجديد ايضا هنا اننا ناخذ البيانات من ثلاثة ملفات نصية و ليس ملف واحد

```
"grab_data('ajax.txt','details')"
"grab_data('css.txt','details')"
"grab_data('php.txt','details')"
```

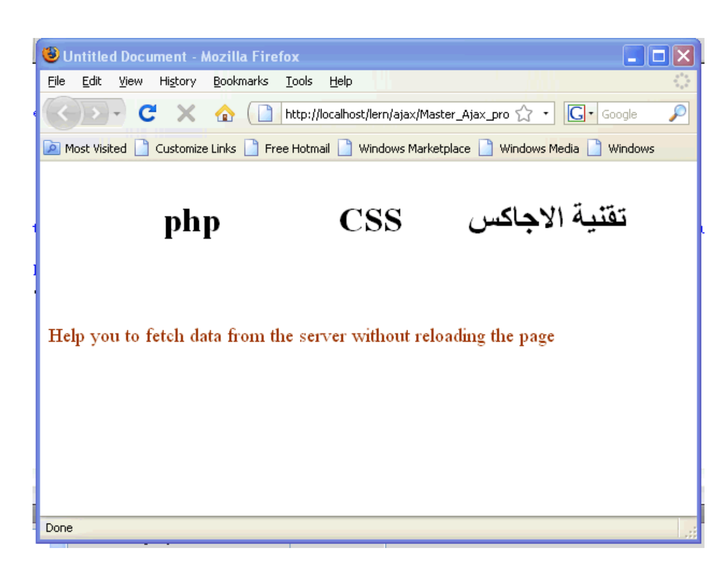
طبعا **details** هو اسم الdiv لاحظ الكود :

```
<div id="details" dir="ltr"></div>
```

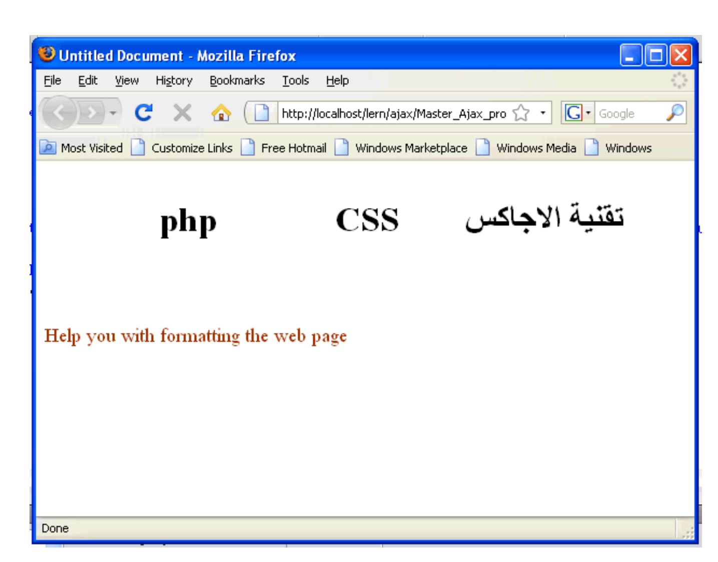
الان لنجربها بالمتصفح:



و اذا وضعت الماوس فوق تقنية الاجاكس ستلاحظ:



و كذلك عند مرور المؤشر فوق CSS سيتغير النص



الاتصال بسكربتات php

تعلمنا جلب بيانات نصية من ملف نصي لكن ماذا لو اردنا الاتصال بسكربت بي اتش بي الامر اسهل مما تتصور فقط غير اسم الباراميتر الثاني المرر للفنكشن ليصبح باسم ملف السكربت اولا انشا ملف جديد بالدريم ويفر او اي محرر و اكتب فيه التالي:

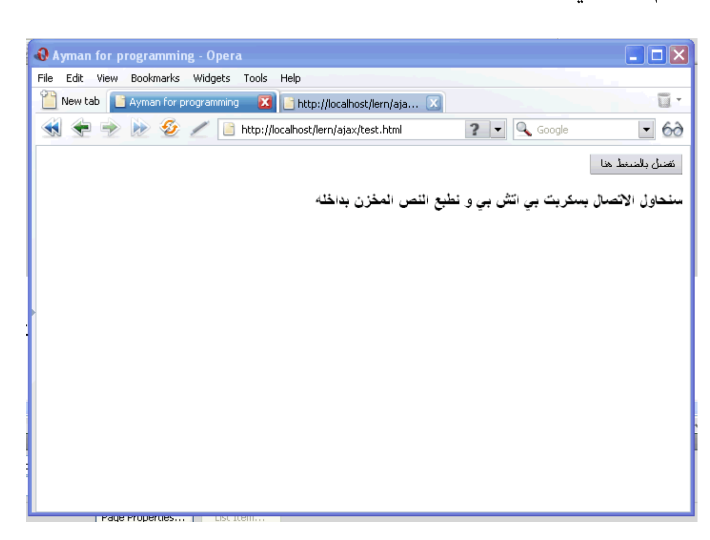
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<?php
print "الحمد لله نجح الاتصال فهذا النص مكتوب بملف بي اتش بي ";
```

```
?>
</body>
</html>
```

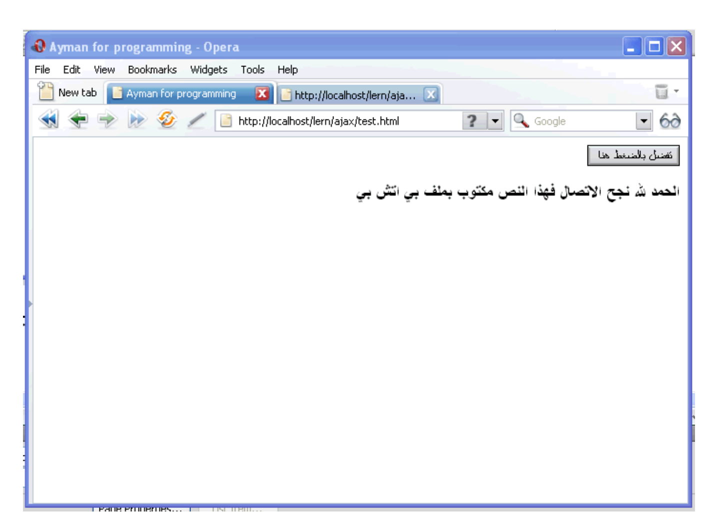
ثم احفظ هذا الملف بالاسم script.php مثلا الان بقي علينا ملف الهوتميل و ليكن test.html اكتب فيه الكود التالي:

```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false;
ajax = new XMLHttpRequest();
//________________
function grab_data(data_sourse,div_id)
{
if(ajax)
{
var ob = document.getElementById(div_id);
ajax.open("get",data_sourse);
ajax.onreadystatechange = function ()
// start new function
{
if(ajax.readyState == 4 && ajax.status == 200)
{
ob.innerHTML = ajax.responseText;
}
}
// End new function
ajax.send(null);
}
}
//________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<form>
<input type="button" onClick="grab_data('script.php','top')" value="تفضل بالضغط هنا">
</form>
<h3>
<div id="top"> سنحاول الاتصال بسكربت بي اتش بي و نطبع النص المخزن بداخله </div>
</h3>
</body>
</html>
```

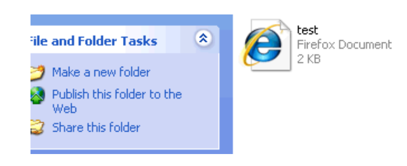
اذا لم نغير شيء عن الامثلة السابقة فقط غيرنا **script.php** الان لنستعرضه في المتصفح اوبرا :



و اذا ضغطنا على الزر يتغير النص :



انتبه الى ضرورة وضع ملف البي اتش بي بنفس المجلد:





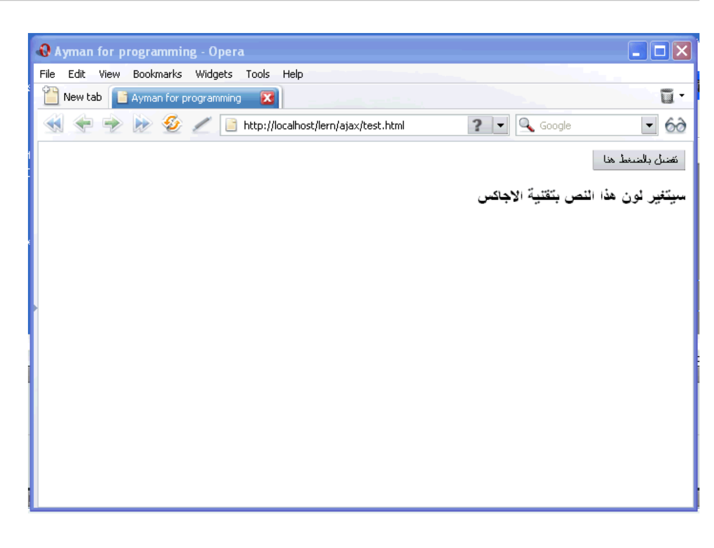
الفصل الثالث تقنيات متقدمة

قبل ان نبدأ بسكربت متقدم ساعطيك امثلة بسيطة و سهلة تمهد لذلك المثال المتقدم اذا لنبدأ ببسم الله

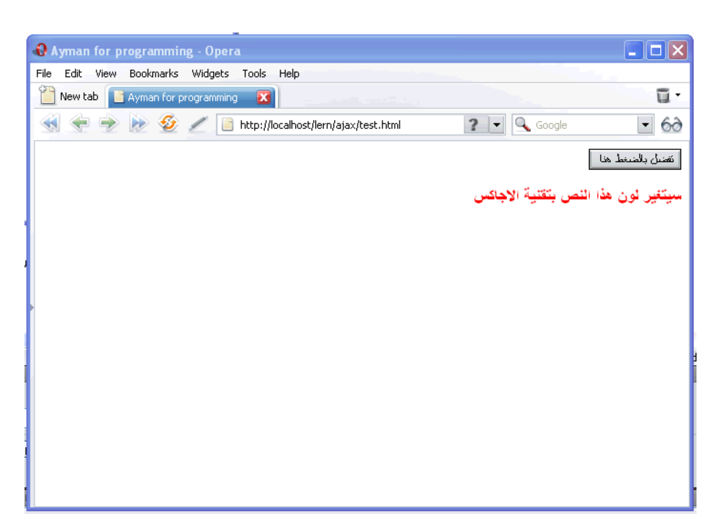
تغير اللون

عندما تكتب رسالة بمحرر نصوص الياهو مثلا تستطيع ان تغير لون النص فورا و بدون تحديث الصفحة يكمن السر باستخدام الخواص style و color لاحظ المثال :

```
<html dir="rtl">
<head>
<script type="text/javascript">
//________________
function change_color(my_color)
{
document.getElementById("top").style.color = my_color;
}
//________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<form>
<input type="button" onClick="change_color('red')" value="تفضل بالضغط هنا">
</form>
<h3>
<div id="top"> سيتغير لون هذا النص بتقنية الاجاكس</div>
</h3>
</body>
</html>
```



و اذا ضغطنا على الزر سيتغير اللون الى الاحمر :



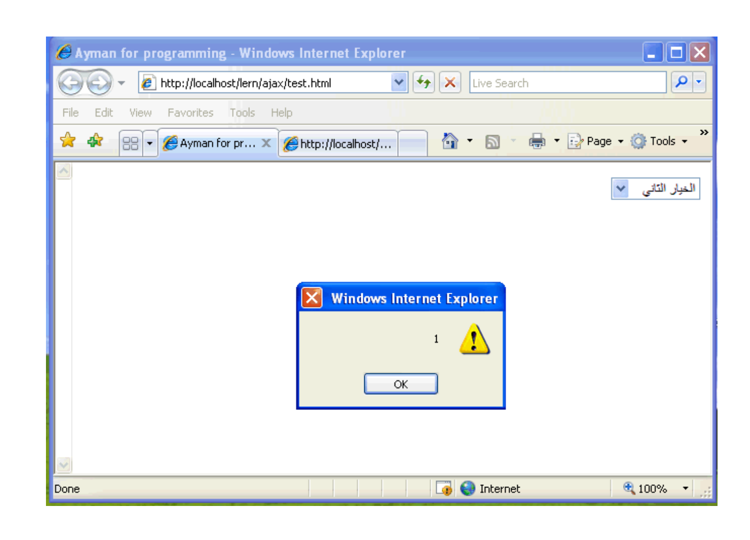
كما ترى اصبح لونه احمر

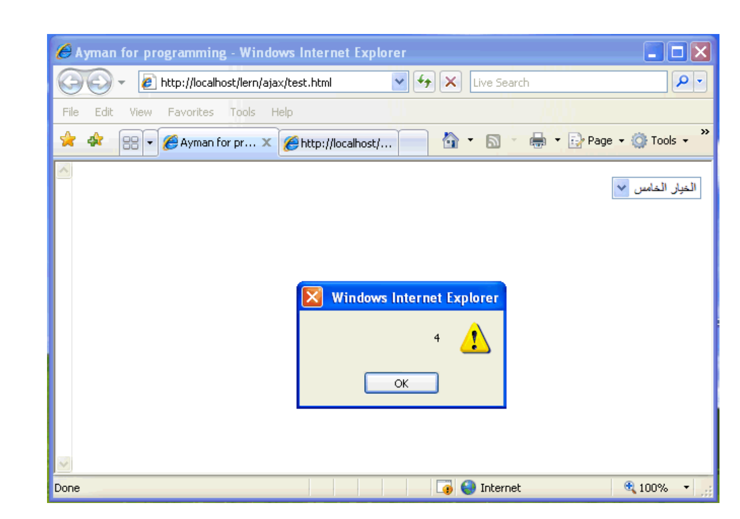
الخاصية selectedIndex

الان سنكتب سكربت بسيط يوضح لك استعمال الخاصية selectedIndex فعندما يكون لديك قائمة منسدلة و يختار المستخدم الخيار الاول تعطيك0 و الخيار الثاني تعطيك 1 و هكذا :

```
<html dir="rtl">
<head>
<script type="text/javascript">
//________________
function show_select()
{
var selected =
document.getElementById("optionList").selectedIndex;
alert(selected);
}
//________________ </script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<form>
<label>
<select id="optionList" onChange="show_select()">
<option >الخيار الاول</option>
<option > الخيار الثاني</option>
<option > الخيار الثالث</option>
<option > الخيار الرابع</option>
<option > الخيار الخامس</option>
</select>
</form>
```

```
</body>
</html>
```





سحب بيانات XML

اولا انشا ملف جديد و اكتب فيه الكود التالي ثم احفظه باسم file.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<options>
<option>red</option>
<option>green</option>
<option>blue</option>
</options>
```

الان نريد جلب الالوان المخزنة لدينا في هذا الملف كيف؟؟ اولا سننشأ فنكشن اسمها get_xml بنفس الطريقة التي تعلمناها سابقا حاول ان تقرأ الكود :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
var options_arr = new Array();
var java_arr = new Array();
//________________
function get_xml(data_sourse,div_id)
{
if(ajax)
{
ajax.open("get",data_sourse);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
xml_document = ajax.responseXML;
options_arr = xml_document.getElementsByTagName("option");
for(var i=0;i<3;i++)
{
java_arr[i]= options_arr[i].firstChild.data;
}
alert(java_arr);
}
}
ajax.send(null);
}
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<form>
<input type="button" onClick="get_xml('file.xml','top')" value="click
here">
</form>
</body>
</html>
```

اولا هذا السطر يستخدم مع المتصفح انترنت اكسبورر :

```
ajax = new ActiveXObject("Microsoft.XMLHTTP");
```

ثم انشأنا مصفوفتين فارغتين سيتم تعبأتهم لاحقا :

var options_arr = new Array();

var java_arr = new Array();

المصفوفة options_arr ستحوي عناصر**XML** و ليس نصوص اما المصفوفة java_arr فهي مصفوفة جافا سكريبت عادية سنخزن فيها عناصر نصية هي الالوان .

اولا سنقوم بجلب بيانات ملف الاكس ام ال و نخزنها بمتغير اسمه **xml_document** و ذلك باستخدام خاصية **responseXML**

```
xml_document = ajax.responseXML;
```

الان سنخزن عناصر الاكس ام ال التي جلبناها في المصفوفة التي عرفناها options_arr و ذلك باستخدام الخاصية getElementsByTagName التي تستدل على العنصر من خلال اسمه

```
xml_document = ajax.responseXML;
options_arr = xml_document.getElementsByTagName("option");
```

الان لدينا المصفوفة options_arr جاهزة و تحوي البيانات لكن صيغة تلك البيانات على شكل عناصر اكس ام ال فكيف سترجمها الى عناصر نصية يكون ذلك بالعبارة التالية:

options_arr[0].**firstChild.data** العنصر الاول

options_arr[1].**firstChild.data** العنصر الثاني

options_arr[2].**firstChild.data** العنصر الثالث

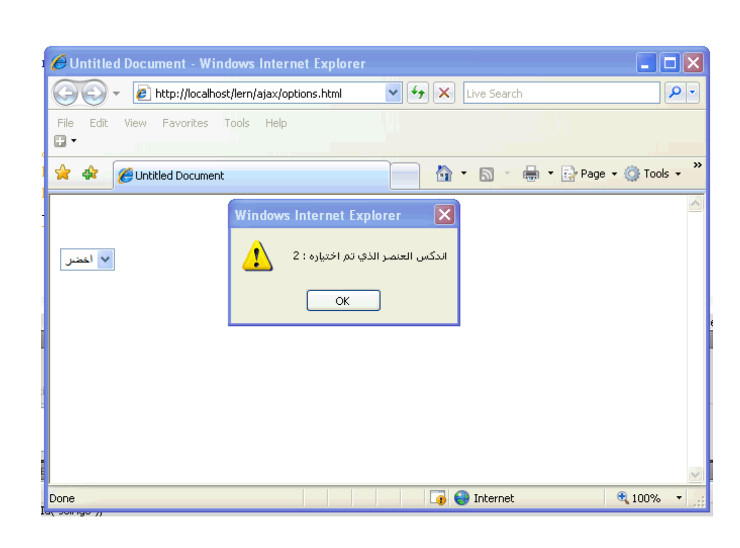
و لكن انا كمبرمج اجاكس لا اكتب هكذا لان هناك اختراع اسمه حلقة فور :

```
options_arr = xml_document.getElementsByTagName("option");
for(var i=0;i<3;i++)
{
java_arr[i]= options_arr[i].firstChild.data;
}
```

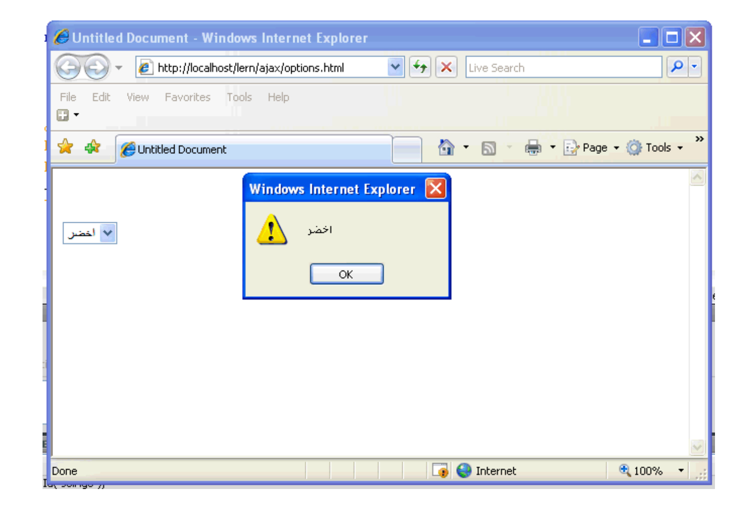
تطبيق اخر

سنقوم بانشاء قائمة اختيارات و حيث ان هناك اكثر من طريقة للتعامل مع النماذج بواسطة الجافا سكريبت حاول قراءة الكود عدة مرات سيكون الشرح داخل الكود :

```
<html>
<head>
<script type="text/javascript">
function ayman_fun()
{
var ob = document.getElementById("optionList");
alert(ob.options[1].firstChild.nodeValue+" : النص");
alert (ob.options[1].text +" : طريقة ثانية للحصول على النص");
//___________________________
alert(ob.options[1].value+" : تعطيك القيمة فاليو");
alert(ob.options[1].getAttribute("value")+" : طريقة اخرى للحصول على الفاليو");
alert(ob.selectedIndex+ " : اندكس العنصر الذي تم اختياره ");
if(ob.options[2].selected) alert("لقد قمت باختيار اللون الاخضر");
var index = ob.selectedIndex ;
var user_select = ob.options[index].text;
alert(user_select);
ob.options[0].text = "new option";
}
//_______________________________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<p> </p>
<form id="form1" name="form1" method="post" action="">
<select id="optionList" onchange="ayman_fun()">
<option value="red">احمر</option>
<option value="blue"> ازرق</option>
<option value="green">اخضر</option>
</select>
</form>
</body>
</html>
```



طبعا عندما تغير اللون سيظهر لك حوالي 5 رسائل و ليس هذه فقط



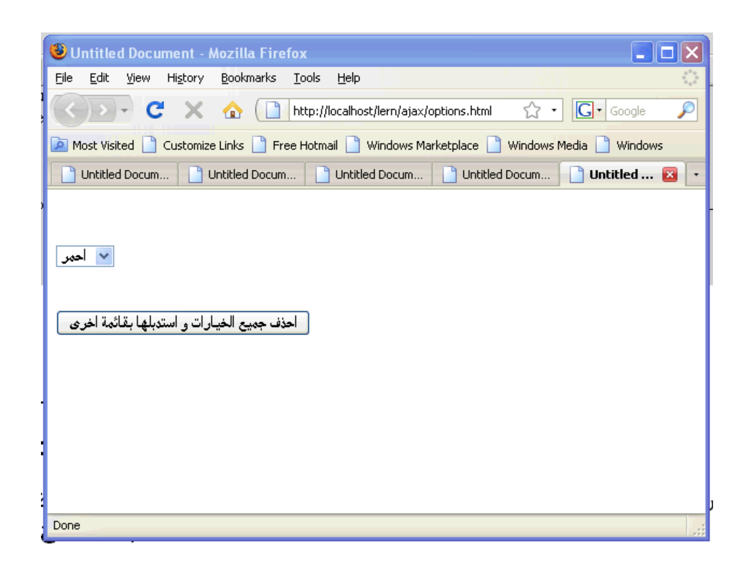
الان جهزت لك تطبيق اجاكس رائع و ان شاء الله تستفيد منه سنسمح للمستخدم بحذف جميع عناصر القائمة و استبدالها بقائمة اخرى:

```
<head>
<script type="text/javascript">
function add_options()
{
var selectControl = document.getElementById('optionList');
selectControl.options.length = 0;
var arr_options = new Array("black","red","green");
for(i=0;i<arr_options.length;i++)
{
selectControl.options[i] = new Option(arr_options[i]);
}
}
//______________________________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<p> </p>
<form id="form1" name="form1" method="post" action="">
<select id="optionList">
<option value="red">احمر</option>
<option value="blue"> ازرق</option>
<option value="green">اخضر</option>
</select>
</form>
<br /><br />
<input type="button" onclick="add_options()" value="احذف جميع الخيارات و استبدلها بقائمة اخرى"/>
</body>
</html>
```

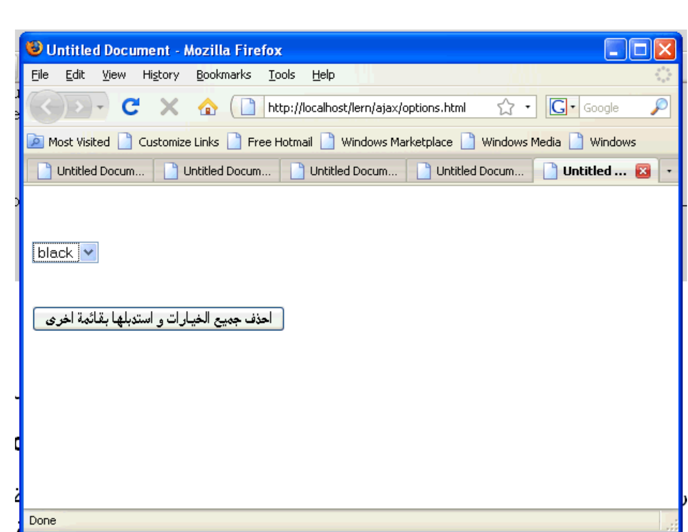
اول فكرة هي حذف جميع عناصر القائمة:

selectControl.options.length = 0;

الفكرة الثانية هي اضافة عناصر جديدة **new Option** الى المصفوفة **options** طبعا حرف **O Option** كبير و ليس **o** صغير و الان لنجربها بالمتصفح :



و بعد الضغط على الزر تصبح القائمة انكليزي :

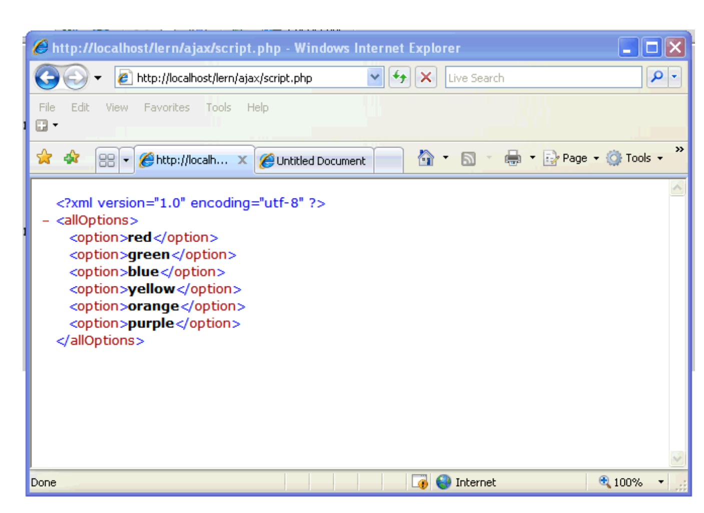


تطبيق متقدم

الان نريد ان نتعلم كيف نسمح للمستخدم بتغيير لون النص كما هو الحال عند كتابتك رسالة بصندوق سحري او محرر نصوص اولا سننشأ ملف البي اتش بي و الذي يحوي بيانات اكس ام ال :

```
<?php
```

```
header("Content-type: text/xml");
print "<?xml version='1.0' encoding='utf-8' ?>";
print "<alloptions>";
$arr = array("red","green","blue");
for($i=0;$i<count($arr);$i++)
{
print "<option>".$arr[$i]."</option>";
}
print "</alloptions>";
?>
```



الان بقي علينا ملف الهوتميل options.html ساكتب الكود كاملا ثم اشرحه شرح مختصر لان معظم الافكار مرت معنا :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//*****************************
var options_arr = new Array();
function getOptions()
{
if(ajax)
{
ajax.open("get","script.php",true);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
xmlDocument = ajax.responseXML;
options_arr = xmlDocument.getElementsByTagName("option");
list_options();
}
}
ajax.send(null);
}
}
//_______________________________________
function setOption()
{
var index =
document.getElementById("optionList").selectedIndex;
document.getElementById("center").style.color =
options_arr[index].firstChild.data;
}
//_________________________________
function list_options()
{
var selectControl = document.getElementById("optionList");
for(i=0;i < options_arr.length;i++)
{
selectControl.options[i] = new
Option(options_arr[i].firstChild.data);
}
}
//_________________________________
</script>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<p> </p>
<form id="form1" name="form1" method="post" action="">
<select id="optionList" onChange="setOption()">
</select>
</form>
<br /><br />
<input type="button" onclick="getOptions()" value="اظهر قائمة الالوان"/>
<h2> غير لون هذا النص: <br>
<div align="center" id="center"> الحمد لله الذي هدانا لهذا و ما كنا لنهتدي لولا ان هدانا
الله</div>
</h2>
</body>
</html>
```



xmlDocument = ajax.responseXML;

اولا سحبنا البيانات من الملف الموجود بالسيرفر باستخدام خاصية responseXML و خزنها في المتغير xmlDocument لكن هذه تمثل الملف باكمله و انا لا اريد سوى عناصر الاوبشن , لسحب عناصر الاوبشن نستخدم الخاصية getElementsByTagName :

options_arr = xmlDocument.getElementsByTagName("option");

و لكن هذه البيانات ما تزال عناصر اكس ام ال و ليست نصوص اي انها على هذا الشكل :

```
<option>red</option>
<option>green</option>
<option>blue</option>
```

لتفريغ النصوص المحتواة داخل المصفوفة لا بد من استخدام الخاصية :firstChild.data

اما لتغيير لون النص نستخدم العبارة
document.getElementById("center").style.color

الفكرة الاخيرة هي بزرع خيارات جديدة او انشا كائنات جديدة من نوع اوبشن و ضمها للمصفوفة : options

```
selectControl.options[i] = new
Option(options_arr[i].firstChild.data);
```

الطريقة GET

غالبا ما نستخدم GET عندما تكون حجم البيانات صغير و ان لا تكون البيانات سرية :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//________________
var optionsArr = new Array();
function getData(colors)
{
if(ajax)
{
var url = "script.php?colors="+colors;
ajax.open("get",url,true);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
var xmlDocument = ajax.responseXML;
optionsArr = xmlDocument.getElementsByTagName("option");
makeOptions();
}
}
ajax.send(null);
}

}
//______________________________________
function makeOptions()
{
selectControl = document.getElementById("selectId");
for(i=0;i < optionsArr.length;i++)
{
selectControl.options[i+1] = new Option(optionsArr[i].firstChild.data) ;
}
}
//______________________________________

//***********************
function setOption()
{
var index = document.getElementById("selectId").selectedIndex;
document.getElementById("textId").style.color =
optionsArr[index].firstChild.data;

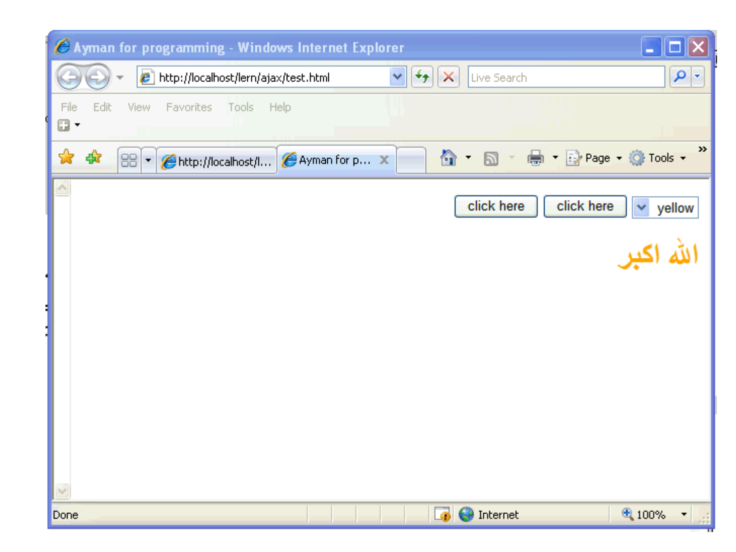
}
//*********************
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<form>
<select id="selectId" onChange="setOption()">
<option selected>اختر لونا </option>
</select>
<input type="button" value="click here" onClick="getData(1)">
<input type="button" value="click here" onClick="getData(2)">
</form>
<div id="textId">
<h1>الله اكبر </h1>
</div>
</body>
</html>
```

عندما تقرأ هذا السطر **var url = "script.php?colors="+colors;**

فذلك يعني اننا نرسل الى السكربت متغير اسمه كولورس و لكن كيف سيقوم ملف البي اتش بي باستقبال هذا المتغير؟؟ الجواب هو باستخدام المصفوفة $_GET[colors] لاحظ كود البي اتش بي :

```
<?php
header("Content-type: text/xml");
print '<?xml version="1.0" encoding="utf-8"?>';
print "<allOptions>";
if($_GET[colors] == 1)
{
$arr = array("red","green","blue");
}
if($_GET[colors] == 2)
{
$arr = array("yellow","orange","purple");
}
for($i=0;$i < count($arr);$i++)
{
print "<option>".$arr[$i]."</option>";
}
print "</allOptions>";
?>
```

يختلف هذا المثال عن الذي قبله اننا اضفنا مجموعة جديدة من الالوان فاذا اختار المستخدم المجموعة الاولى سيرسل متغير colors بواسطة جيت الى السكربت و هذا المتغير سيحمل القيمة واحد اما اذا ضغط المستخدم الزر الثاني سيحمل المتغير القيمة 2:



حيث ان اختيارات الالوان لن تظهر الا اذا ضغطت على احد الزرين

الطريقة POST

هذه الطريقة نستخدمها عندما نريد لإرسال بيانات ضخمة و سرية حيث يتم تشفيرها لكي لا يتمكن الهكر من التقاطها :

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript">
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//_____________________________________
var optionsArr = new Array();
function getData(colors)
{
if(ajax)
{
var url = "script.php";
ajax.open("post",url,true);
ajax.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
xmlDocument = ajax.responseXML;
optionsArr = xmlDocument.getElementsByTagName("option");
makeOptions();
}

}
ajax.send("colors="+colors);
}
}
//______________________________
function makeOptions()
{
var selectControl = document.getElementById("selectId");
for(var i =0;i<optionsArr.length;i++)
{
selectControl.options[i] = new Option(optionsArr[i].firstChild.data);
}
}
//______________________________
//
//@@@@@@@@@@@@@@@@@@@@@@@@
function setOption()
{
var index = document.getElementById("selectId").selectedIndex;
document.getElementById("center").style.color =
optionsArr[index].firstChild.data
}
//@@@@@@@@@@@@@@@@@@@@@@@@
</script>
</head>
<body>
<form>
<select id="selectId" onchange="setOption()">
<option> choos one </option>
</select><br />
<input type="button" value="مجموعة الالوان الاولى" onclick="getData(1)" />
<input type="button" value=" مجموعة الالوان الثانية" onclick="getData(2)"
/></form>
<div align="center" id="center">
<h1>صلى الله على محمد صلى الله عليه و سلم </h1>
</div>
</body>
</html>
```

اولا بعد ان استخدمنا الخاصية اوبن و مررنا لها القيمة بوست الان يجب ان نضع الهيدر للصفحة و

يكون ذلك باستخدام الخاصية setRequestHeader:

```
ajax.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
```

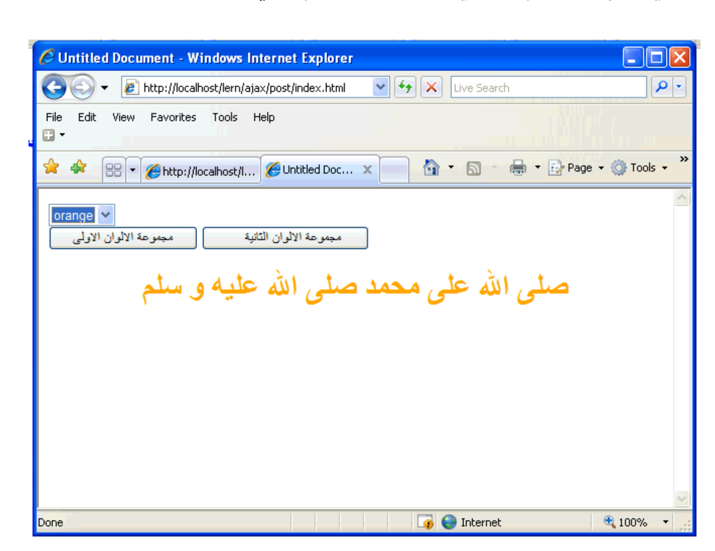
و هناك فكرة جديدة ايضا هي انه مررنا قيمة للخاصية سيند عوضا عن القيمة نال التي تعودنا عليها عندما كنا نستخدم جيت:

```
ajax.send("colors="+colors);
```

فالسطر السابق يعني اننا نرسل المتغير كولورس و قيمة ذلك المتغير و تذكر انه عندما تستخدم الطريقة جيت عندها تمرر القيمة نال null و الان الى سكربت البي اتش بي :

```
<?php
header("Content-type:text/xml");
print '<?xml version="1.0" encoding="utf-8"?>';
print "<allOptions>";
if($_POST[colors] == 1)
{
$arr = array("red","green","blue");
}
if($_POST[colors] == 2)
{
$arr = array("yellow","orange","purple");
}
for($i=0;$i<count($arr);$i++)
{
print "<option>";
print $arr[$i];
print "</option>";
}
print "</allOptions>";
?>
```

الجديد هو اننا استبدلنا جيت GET_$ بالمتغير **POST_$** فقط . لنجربها بالمتصفح:



جافا سكربت من جهة السيرفر

احيانا تضطر الى تنفيذ كود جافا سكربت من السيرفر اي ان كود الجافا سكربت يكون مكتوب بملف البي اتش بي و تقوم انت بسحبه بنفس الطريقة التي تسحب بها نص عادي :

```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false;
ajax = new XMLHttpRequest();
//_______________
function grab_data(data_sourse)
{
if(ajax)
{
ajax.open("get",data_sourse);
ajax.onreadystatechange = function ()
// start new function
{
if(ajax.readyState == 4 && ajax.status == 200)
{
eval(ajax.responseText);
}
}
// End new function
ajax.send(null);
}
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<form>
<input type="button" onClick="grab_data('java.php')" value="Get java
from the server"><h1>تقنية الاجاكس </h1>
</form>
<p> </p><font size="4" color="#993300">
</font>
</body>
</html>
```

الجديد هو هذا السطر **eval(ajax.responseText);**

فاذا كان النص الذي جلبته عبارة عن كود جافا سكريبت و تريد تنفيذ هذا الكود استخدم الخاصية

و eval الان الى ملف البي اتش بي java.php :

```
<?php
print "alert('هذا كود جافا سكربت تم جلبه من السيرفر ')";
?>
```

كما ترى بسيط جدا و هذه المرة لا يحوي نصا او بيانات اكس ام ال و انما يحوي كود جافا سكربت و الان لنستعرضه



و لكن انتبه لامر خطير و هو انك لن تستطيع كتابة كود طويل لاكثر من سطر و اذا فعلت فلن يتم تنفيذه لذلك نحن غالبا ما نستخدم جافا سكربت القادمة من السيرفر لاستدعاء فنكشن فقط و هذا المثال يوضح :

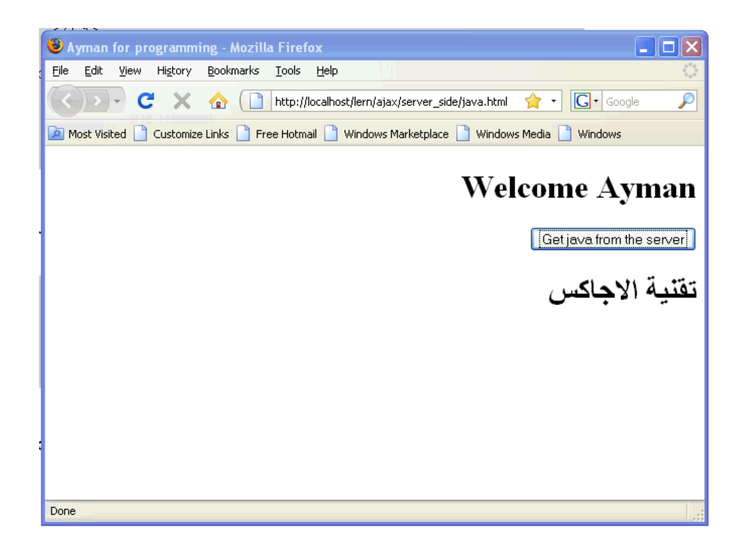
```
<html dir="rtl">
<head>
<script type="text/javascript">
var ajax = false;
ajax = new XMLHttpRequest();
//________________
function grab_data(data_sourse)
{
if(ajax)
{
ajax.open("get",data_sourse);
ajax.onreadystatechange = function ()
// start new function
{
if(ajax.readyState == 4 && ajax.status == 200)
{
//ob.innerHTML = ajax.responseText;
eval(ajax.responseText);
}
}
// End new function
ajax.send(null);
}
}
//____________
function hi(name)
{
document.getElementById("top").innerHTML = "<h1>Welcome " + name + "</h1>";
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<div id="top" > هنا سيتم تنفيذ الفنكشن ان شاء الله </div>
<form>
<input type="button" onClick="grab_data('java.php')" value="Get java from the server"><h1>تقنية الاجاكس </h1>
</form>
<p> </p><font size="4" color="#993300">
</font>
</body>
</html>
```

و الان لننتقل الى ملف البي اتش بي الذي سيستدعي الفنكشن:

```
<?php
print "hi(' Ayman ')";
?>
```

طبعا اذا كتبت اكثر من سطر لن يعمل الكود مطلقا و الان لنجربه بالمتصفح :

و عندما تضغط على الزر سيتم استدعاء الفنكشن بأمر قادم من السيرفر :



استرجاع كائن جافا سكريبت

الان لدينا كائن جافا سكريبت سنكتبه على صيغة نص و ليتم تحويله الى هيئة كائن لاحقا هذا الكائن ستكون مهمته استدعاء هذه الفنكشن :

```
function math(op1, op2)
{
var sum = op1 + op2;
alert(op1 + " + " + op2 + " = " + sum);
}
```

طبعا نحن نستدعي هذه الفنكشن بالطريقة التقليدية هكذا :

```
math(5,8);
```

و لكن هذه المرة سنقوم بانشاء كائن لديه ثلاث خواص:

اسم الفنكشن مثلا : math

المعامل الاول مثلا 5

و المعامل الثاني مثلا 8

الان لدينا متغير نصي تعمدنا كتابته بطريقة عجيبة او غير مألوفة بالنسبة لك لكي نستطيع تحويله الى كائن و بالتالي يقوم الكائن باستدعاء الفنكشن :

```
var text = "{method: 'math' , num1: 24 , num2: 6};";
```

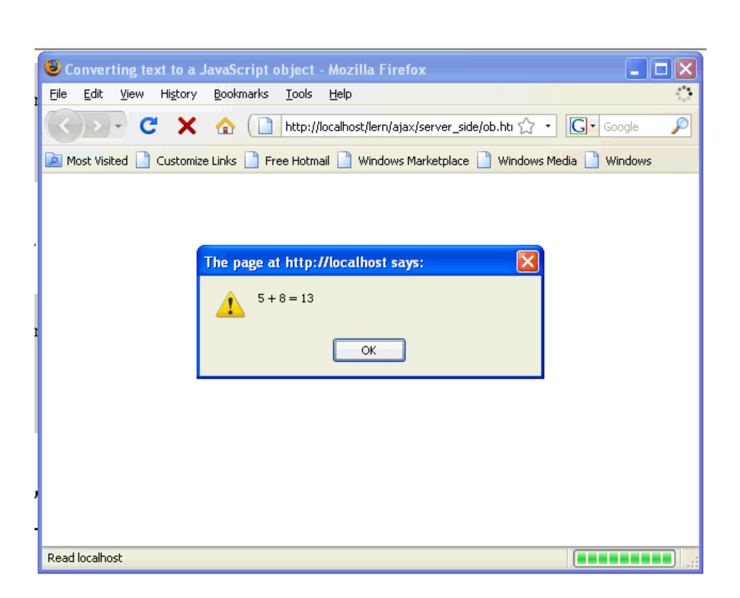
الان تستطيع انشاء كائن **ob** من هذا النص طالما ان النص يحتوي على الخواص الثلاثة التي ستلزمك :

```
var text = "{method: 'math' , num1: 24 , num2: 6};";
var ob ;
eval('ob='+text);
```

لقد استخدمنا الدالة **eval** لانشاء الكائن الان لم يبق علينا الا ان نستدعي الفنكشن :

```
var text = "{method: 'math' , num1: 5 , num2: 8};";
var ob ;
eval('ob='+text);
eval(ob.method + '(' + ob.num1 + ',' + ob.num2 + ');');
```

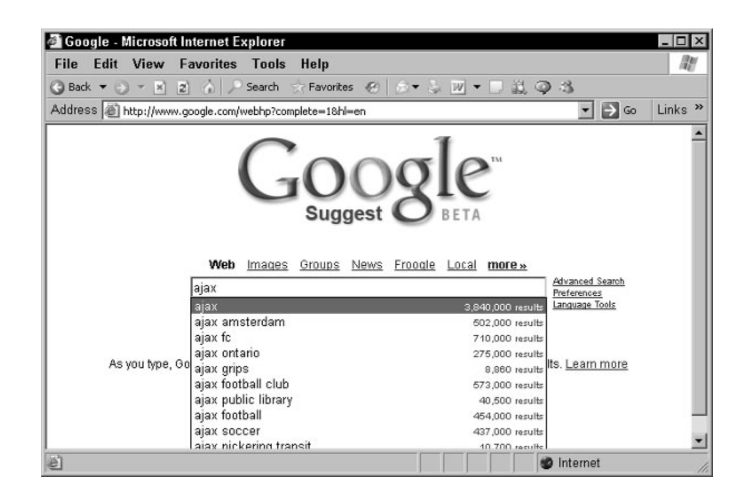
نحن بذلك نكون قد استدعينا الفنكشن ماث حيث كتبنا الخاصية التي تعبر عن اسم الفنكشن ثم فتحنا قوس و كتبنا الخاصيتين اللتان تعبران عن الارقام ثم اغلقنا القوس



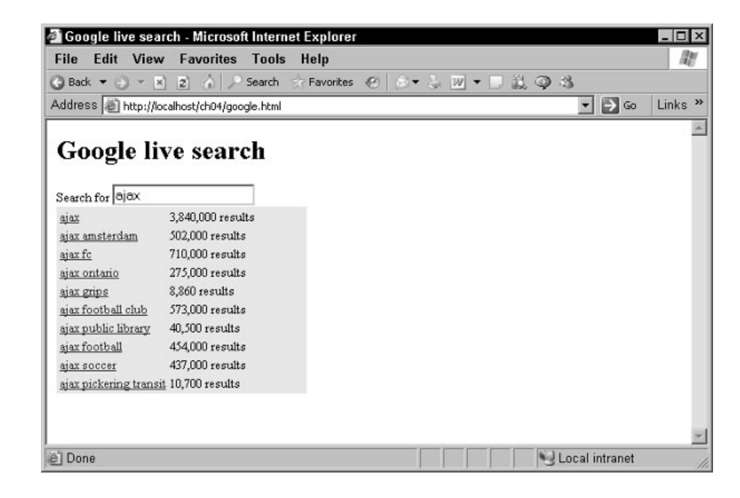
google suggest

عندما تكتب في حقل النص بغوغل كلمة دروس مثلا تخرج قائمة منسدلة تقترح عليك اقتراحات مثلا

دروس الفوتوشوب , دروس رياضيات وهكذا و هذا ما يسمى بـ google suggest



طبعا يمكنك ان تستخدم نفس التقنية بموقعك حيث تتصل بسيرفر غوغل بدون اي تحديث للصفحة :



حيث تلقينا من غوغل سطر جافا سكريبت يقوم هذا السطر باستدعاء الفنكشن sendRPCDone و هذه هي البراميترات :

```
sendRPCDone(unusedVariable, searchTerm, arrayTerm, arrayResults,
unusedArray)
```

معالجة البيانات القادمة من غوغل

اولا سننشئ حقل نص و سنربطه بالفنكشن getSuggest عن طريق الحدث onkeyup هذا الحدث يتحقق بمجرد ان يكتب المستخدم او يمسح اي حرف :

```
البحث عن <input type="text" name="textfield" id="textfield"
onKeyUp="getSuggest(event)">
```

الان سنقوم بانشاء عنصر div لتظهر فيه الاقتراحات :

```
<div id="divId"></div><div></div>
```

لنقم ببناء الفنكشن getSuggest ثم نمرر لها الكائن keyEvent لكن هذا الكائن غير مدعوم من قبل الانترنت اكسبلورر في هذا المتصفح عليك استخدام window.event :

```
function getSuggest(keyEvent)
{
keyEvent = (keyEvent) ? keyEvent: window.event;
-
-
}
```

يمكنك ايضا ان تحدد اين يكتب المستخدم لكن هذا يعتمد على نوع المتصفح بالفايرفوكس استخدم الخاصية target اما بالانترنت اكسبلورر استخدم srcElement:

```
function getSuggest(keyEvent)
{
keyEvent = (keyEvent) ? keyEvent: window.event;
input = (keyEvent.target) ? keyEvent.target : keyEvent.srcElement;
-
-
}
```

و يمكنك ان تحدد نوع الكائن keyEvent هل هو من نوع keyup:

```
function getSuggest(keyEvent)
{
keyEvent = (keyEvent) ? keyEvent: window.event;
input = (keyEvent.target) ? keyEvent.target : keyEvent.srcElement;
if (keyEvent.type == "keyup") {
-
-
}
}
```

الاتصال بغوغل

سنتحقق هل يكتب المستخدم شيئا؟ نتصل بغوغل و نعرض الاقتراحات :

```
function getSuggest(keyEvent)
{
keyEvent = (keyEvent) ? keyEvent: window.event;
input = (keyEvent.target) ? keyEvent.target : keyEvent.srcElement;
if (keyEvent.type == "keyup") {
if (input.value) {
getData("google.php?qu=" + input.value);
}
else {
-
-
}
}
}
```

العبارة else تعني هنا ان المستخدم قد مسح الكلمة و بالتالي نحن بدورنا سنمسح الاقتراحات :

```
function getSuggest(keyEvent)
{
keyEvent = (keyEvent) ? keyEvent: window.event;
input = (keyEvent.target) ? keyEvent.target : keyEvent.srcElement;
if (keyEvent.type == "keyup") {
if (input.value) {
getData("google.php?qu=" + input.value);
}
else {
var targetDiv = document.getElementById("targetDiv");
targetDiv.innerHTML = "<div></div>";
}
}
}
```

لننتقل الى الفنكشن getData التي نمرر لها سكربت البي اتش بي و الذي يقوم بدوره بالاتصال بغوغل اعتقد انك قد حفظت هذه الفنكشن لانها تكررت مكعنا كثيرا

```
if (window.ActiveXObject)
{
ajax = new ActiveXObject("microsoft.xmlhttp");
} else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//_____________________________
function getData(dataSourse)
{
alert("fun");
if(ajax)
{
ajax.open("get",dataSourse);
ajax.onreadystatechange = function ()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
eval(ajax.responseText);

}
}
ajax.send(null);
}
}
```

استخدمنا الفنكشن الجاهزة eval لتنفيذ كود الجافا سكريبت و الان لم يبق علينا الا الفنكشن sendRPCDone التي سنقوم ببنائها سويا :

```
function sendRPCDone(unusedVariable, searchTerm, arrayTerm,
arrayResults, unusedArray)
{
.
.
.
}
```

نحتاج لان نعرف متغير اسمه داتا يحوي عناصر المصفوفتين بشكل منسق وفق جدول حيث يمكننا طباعة عناصر المصفوفة بحلقة تكرار فور :

```
function sendRPCdone
(unusedvariable,searchTerm,arrayTerm,arrayResults,unusedArray)
{
var data = "<table>";
var i ;
if(arrayResults.length != 0)
{
for(i=0;i < arrayResults.length;i++)
{
data += "<tr><td>";
data += "<a href = 'http://www.google.com?
q="+arrayTerm[i]+"'>";
data += arrayTerm[i]
data += "</a> </td><td>"
data += arrayResults[i];
data += "</td></tr>";
}
}
data += "</table>";
document.getElementById("divId").innerHTML = data;
}
```

كما ترى وضعنا رابط بحيث اذا ضغط المستخدم على احد هذه الاقتراحات سيتم البحث عنه و اذا اردت تنسيق تلك الاقتراحات استخدم السي اس اس :

```
<style type="text/css">
#divId {
background-color: #FFEEAA;
width: 30%;
</style>
```

الان ساقدم لك الكود كاملا :

```
<html dir="rtl">
<head>
```

```
<style type="text/css">
#divId{
background-color:#0CF;
width:40%
}
</style>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript">
if (window.ActiveXObject)
{
ajax = new ActiveXObject("microsoft.xmlhttp");
}

else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//______________________________
function getData(dataSourse)
{
if(ajax)
{
ajax.open("get",dataSourse);
ajax.onreadystatechange = function ()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
eval(ajax.responseText);

}
}
ajax.send(null);
}
}
//______________________________
// getSuggest
//**********************
function getSuggest(keyEvent)
{
keyEvent = (keyEvent) ? keyEvent: window.event;
input = (keyEvent.target) ? keyEvent.target :
keyEvent.srcElement;
if (keyEvent.type == "keyup") {
if (input.value) {
getData("google.php?qu=" +
input.value);
}
else {
var targetDiv = document.getElementById("targetDiv");
targetDiv.innerHTML = "<div></div>";
}
}
}//**********************
//sendRPCdone
//++++++++++++++++
function sendRPCdone
(unusedvariable,searchTerm,arrayTerm,arrayResults,unusedArray)
{
var data = "<table>";
var i ;
if(arrayResults.length != 0)
{
for(i=0;i < arrayResults.length;i++)
{
data += "<tr><td>";
data += "<a href = 'http://www.google.com?q="+arrayTerm[i]+"'>";
data += arrayTerm[i]
data += "</a> </td><td>"
data += arrayResults[i];
data += "</td></tr>";
}
}
data += "</table>";
document.getElementById("divId").innerHTML = data;
}
//++++++++++++++++
</script>
</head>
<body>
البحث عن <input type="text" name="textfield" id="textfield"
onKeyUp="getSuggest(event)">
<div id="divId"></div><div></div>
</body>
</html>
```

و اخيرا انتهينا الحمد لله من ملف الهوتميل بقي علينا ملف البي اتش بي google.php و هو بسيط جدا يعتمد على الدالة fopen:

```
<?php
$handle = fopen("http://www.google.com/complete/search?
hl=en&js=true&qu=" .$_GET["qu"], "r");
while (!feof($handle)){
$text = fgets($handle);
echo $text;
}
fclose($handle);
?>
```

الدالة feof تعطي القيمة true اذا وصلنا الى نهاية الملف اما الدالة fgets فتساعدنا على قراءة البيانات و تخزينها في متغير نصي $text و اخيرا نقوم باغلاق الملف بالدالة fclose لتحرير الذاكرة .

و قد تتسائل لماذا لم نستخدم ملف واحد فقط بدلا من ملفين بمعنى لماذا لم نتصل بغوغل مباشرة من ملف الهوتميل دون الحاجة لسكربت البي اتش بي و الجواب هو لتفادي رسائل التحذير المتكررة التي يظهرها المتصفح لدواعي امنية .

التحقق عبر السيرفر

طبعا نحن تعودنا ان يتم التحقق من النماذج بالجافا سكريبت مباشرة اي بالمتصفح دون الحاجة الى الرجوع الى السيرفر لكن ماذا لو اردت التحقق من كلمة مرور ؟ فهل ستضع كلمات المرور ضمن كود الجافا سكريبت و تقدمها على طبق من ذهب للهكر؟!!! طبعا لا الحل سهل جدا تقوم بانشاء فنكشن checkName و تربطها بالحدث onKeyUp فاذا كتب المستخدم اسم محجوز تظهر له رسالة عذرا فهذا الاسم محجوز:

```
<html dir="rtl">
<head>
<style type="text/css">
#top{
width:150px;
background-color:#0FF;
color:#F00;
font-size:18px;
font-weight:900;
}
</style>
<script type="text/javascript">
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("microsoft.xmlhttp");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//______________________
function getData(divId,dataSourse)
{
if(ajax)
{
ajax.open("get",dataSourse);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
document.getElementById(divId).innerHTML = ajax.responseText;
}
}
ajax.send(null);
}
}

//_____________________
//
//*************
function checkName(keyEvent)
{
keyEvent = (keyEvent) ? keyEvent : window.event;
input = (keyEvent) ? keyEvent.target : keyEvent.srcElement
if(keyEvent.type == "keyup")
{
if(input.value)
{
getData("top","login.php?user_name="+input.value);
}
else
{
document.getElementById("top").innerHTML = "<div></div>";
}
}
}
//*************
</script>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<input type="text" name="user_name" id="user_name"
onKeyUp="checkName(event)">
<div id="top"></div>
</body>
</html>
```

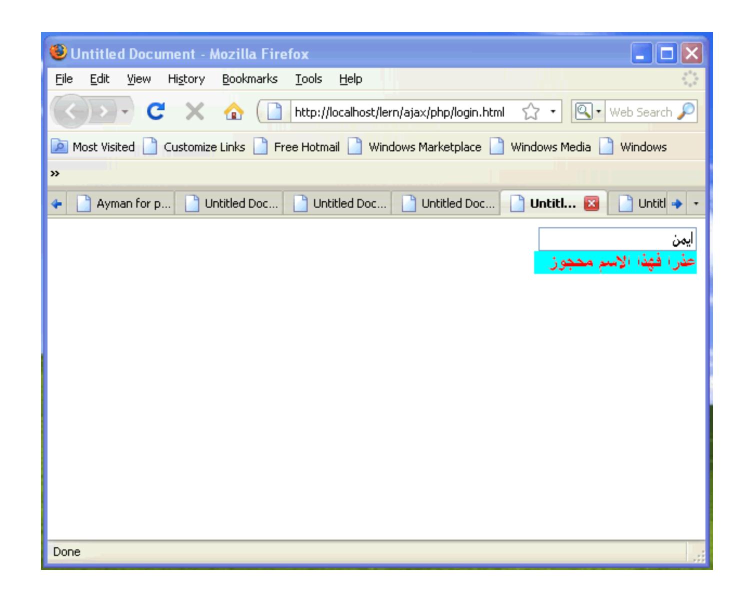
لا شيئ جديد نتحقق من ان المستخدم يكتب في الحقل :

if(input.value)
{
getData("top","login.php?user_name="+input.value);
}

سنرسل ما يكتبه في متغير اسمه يوزر نيم لسكربت البي اتش بي **login.php** :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<?php
extract($_GET);
if($user_name == "ayman" || $user_name == "ايمن" || $user_name
== "ايمن كزة")
{
print "عذرا فهذا الاسم محجوز";
}
?>
</body>
</html>
```

كما ترى وضعنا 3 اسماء محظورة لكن بالعادة نستعلم من قواعد البيانات و الان لنجربها بالمتصفح :



الحلو بالموضوع انك تتحقق عبر السيرفر بدون تحديث الصفحة و هو ما يدل على انك مبرمج اجاكس

محترف لكن هناك ملاحظة ان هذا قد يؤدي الى ارهاق السيرفر

الحصول على معلومات الهيدر

نحن تعودنا على استخدام احد الطريقتين جيت او بوست مع الخاصية open :

;(ajax.open("get",dataSourse

لكن ماذا لو اردنا الحصول على معلومات حول ذلك الملف و حول السيرفر و ليس طباعة محتويات الملف مثلا نريد ان نعرف متى تم تعديل الملف ؟ في هذه الحالة نستخدم الطريقة HEAD اي بهذه الطريقة :

;(ajax.open("HEAD",dataSourse

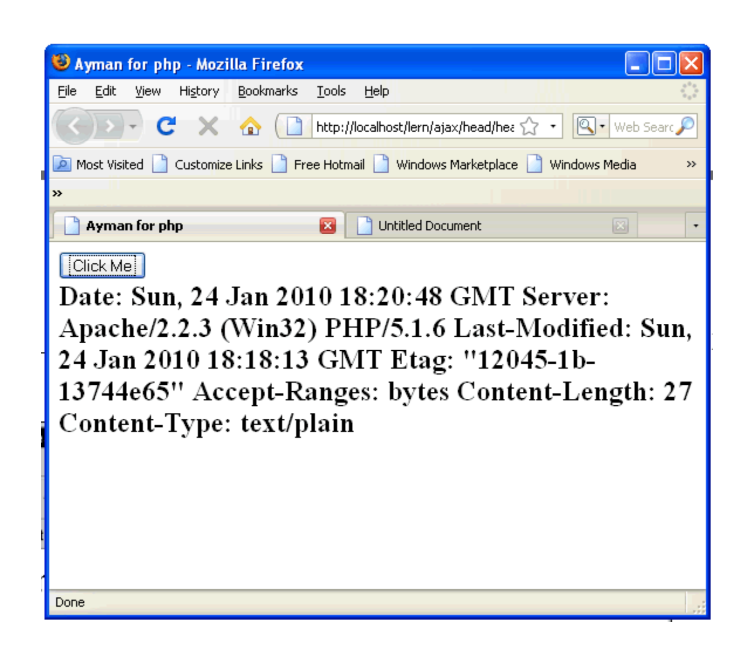
و بالمثال يتضح المقال :

```
<html>
<head>
<script type="text/javascript">
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//_______________
function getData(dataSourse,divId)
{
if(ajax)
{
var ob = document.getElementById(divId);
ajax.open("HEAD",dataSourse);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
ob.innerHTML = ajax.getAllResponseHeaders();
}
}
ajax.send(null);
}
}
//___________________________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for php</title>
</head>

<body>
<input type="button" value="Click Me"
onclick="getData('data.txt','centerDiv')" />
<b><big>
<div id="centerDiv"></div>
</big></b>
</body>
</html>
```

الفكرة الثانية الجديدة هنا هي اننا استخدما الخاصية **getAllResponseHeaders** عوضا عن الخاصية responseText و السبب اننا لا نريد محتويات الملف النصي و انما نريد الهيدر.

و الان لنستعرضها بالمتصفح:

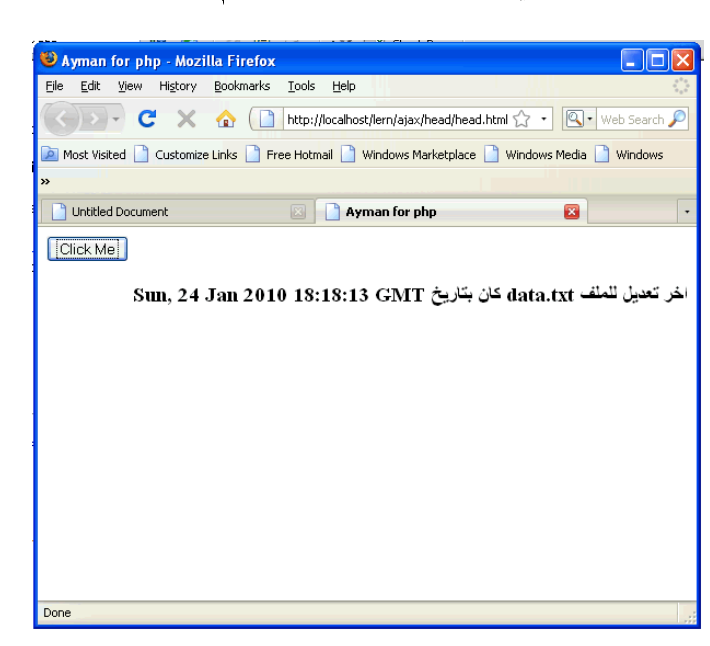


رائع جدا هذه جميع محتويات الهيدر لكن ماذا لو اردت جزء معين من الهيدر مثلا نريد استخلاص اخر تعديل **Last-Modified** و الحل سهل جدا نستخدم getResponseHeader عوضا عن getAllResponseHeaders اي انك لا تريد كل محتوى الهيدر :

```
<html>
<head>
<script type="text/javascript">
var ajax = false;
```

```
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//________________
function getData(dataSourse,divId)
{
if(ajax)
{
var ob = document.getElementById(divId);
ajax.open("HEAD",dataSourse);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
ob.innerHTML = " اخر تعديل للملف data.txt كان بتاريخ ";
ob.innerHTML += ajax.getResponseHeader("Last-Modified");
}
}
ajax.send(null);
}
}
//____________________________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for php</title>
</head>
<body>
<input type="button" value="Click Me"
onclick="getData('data.txt','centerDiv')" />
<h3>
<div dir="rtl" id="centerDiv"></div>
</h3>
</body>
</html>
```

كما ترى مررنا لها قيمة **Last-Modified** ولم نتركها فارغة و الان لنراها في المتصفح :



طبعا الهيدر التي جلبتها من السيرفر على صيغة نص ما رايك ان نحولها الى كائن تاريخ Date لكي يمكنك المقارنة بين تاريخين مثلا :

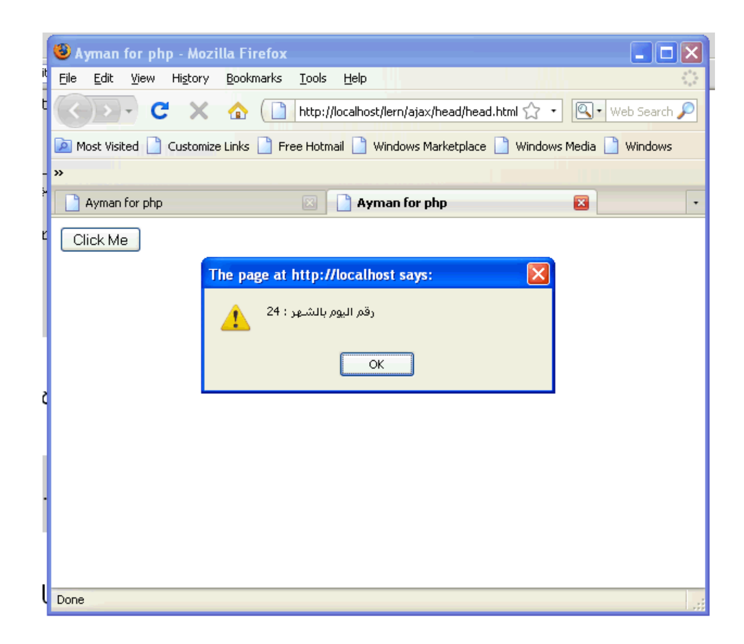
```
<html>
<head>
<script type="text/javascript">
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//________________
function getData(dataSourse,divId)
{
if(ajax)
{
var ob = document.getElementById(divId);
ajax.open("HEAD",dataSourse);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
var date = new Date(ajax.getResponseHeader("Last-Modified"));
alert (" رقم اليوم بالشهر : " + date.getDate());
alert ("رقم اليوم بالاسبوع من واحد الى ستة: " + date.getDay());
alert ("رقم الشهر بالسنة من صفر الى احدى عشر: " + date.getMonth());
alert (" السنة : " + date.getFullYear());
alert (" الساعة: " + date.getHours());
alert (" الدقيقة: " + date.getMinutes());
alert (" الثانية : " + date.getSeconds());
}
}
ajax.send(null);
}
}
//____________________________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for php</title>
</head>
<body>
<input type="button" value="Click Me"
onclick="getData('data.txt','centerDiv')" />
<h3>
<div dir="rtl" id="centerDiv"></div>
</h3>
</body>
</html>
```

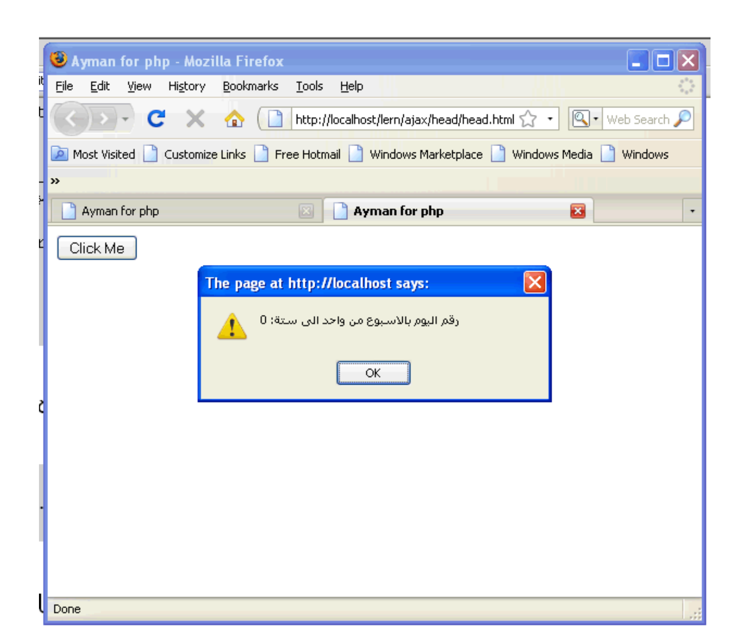
عرفنا متغير جديد اسميناه ثم عرفناه على انه كائن تاريخ:

(("var date = new Date(ajax.getResponseHeader("Last-Modified

و طالما لديك الكائن يمكنك الاستفادة من جميع خواص هذا الكائن مثلا الخاصية **getHours** تعطيك الساعة . . .

و سيقوم المتصفح بعرض 6 رسائل :



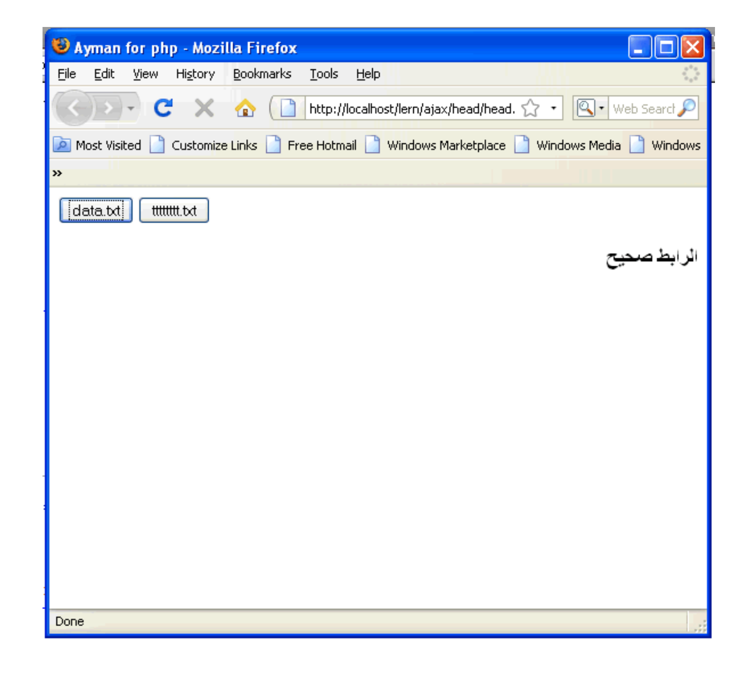


التحقق من رابط

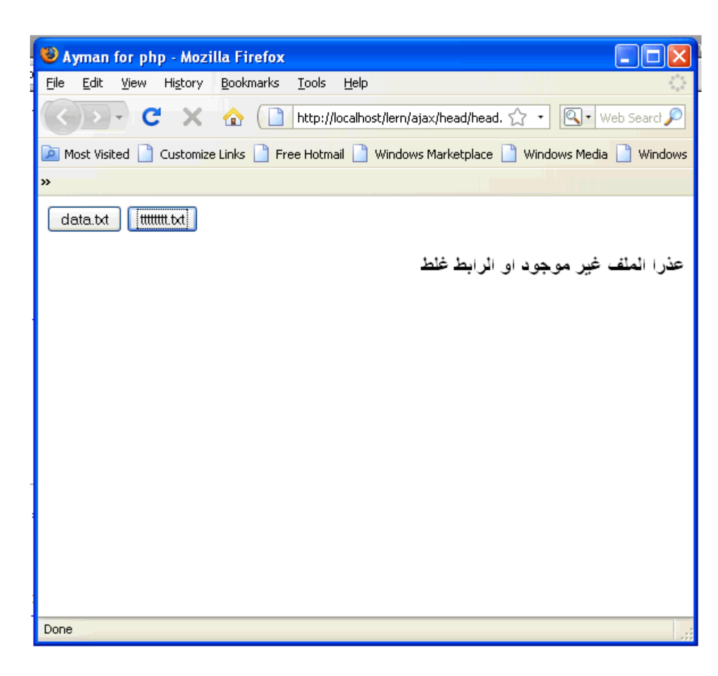
قبل ان نحمل ملف يمكننا ان نتحقق من صحة الرابط و يكون ذلك بوضع شرط اذا اكنت الخاصية status تساوي 200 فهذا يعني ان كل شيئ تمام و الرابط صحيح اما اذا اكنت status تحمل القيمة 404 فذلك يعني ان الرابط خطأ :

```
<html>
<head>
<script type="text/javascript">
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
//_______________
function getData(dataSourse,divId)
{
if(ajax)
{
var ob = document.getElementById(divId);
ajax.open("HEAD",dataSourse);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 )
{
if(ajax.status == 200)
{
ob.innerHTML = "الرابط صحيح ";
}
else if (ajax.status == 404)
{
ob.innerHTML = " عذرا الملف غير موجود او الرابط غلط";
}
}
}
ajax.send(null);
}
}
//__________________________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for php</title>
</head>
<body>
<form>
<input type="button" onClick="getData('data.txt')"
value="data.txt">
<input type="button" onClick="getData('tttttt.txt')"
value="tttttttt.txt">
</form>
<h3>
<div dir="rtl" id="centerDiv"></div>
</h3>
</body>
</html>
```

فهذا الكود مفيدة جدا فعند فحصك للملف **data.txt** ستظهر لك رسالة بالمتصفح :



اما اذا حاولت التحقق من الملف **tttttttt.txt** طبعا هذا الملف غير موجود :



اكثر من كائن **XMLHttpRequest**

ماذا لو كان لديك عدة ازرار و قام المستخدم بالضغط عليها واحد تلو الاخر و بسرعة جدا فكثير من المستخدمين يفعلون ذلك , طبعا هذا سيؤدي الى تشويش السكربت و ارباكه لان المستخدم يرسل طلب جديد قبل ان تكتمل الاستجابة للطلب القديم !!

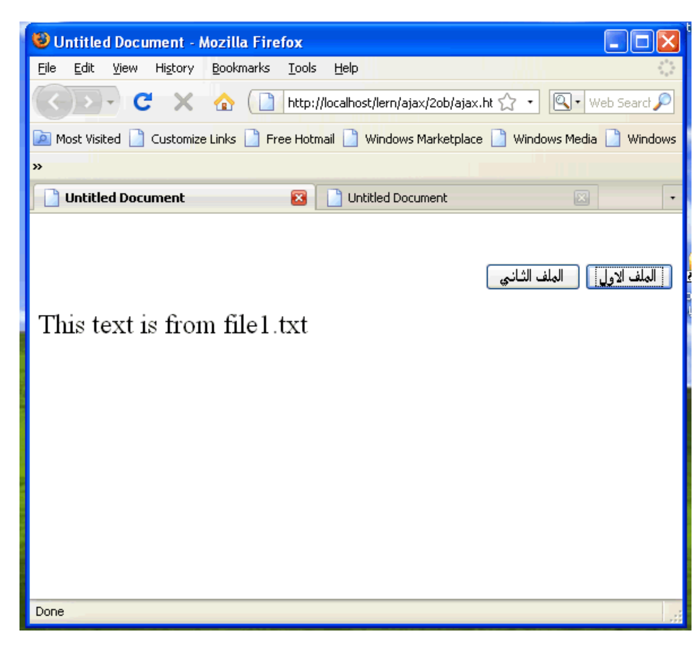
و ياتي الحل هنا بانشاء كائنات متعددة XMLHttpRequest بنفس الطرق السابقة :

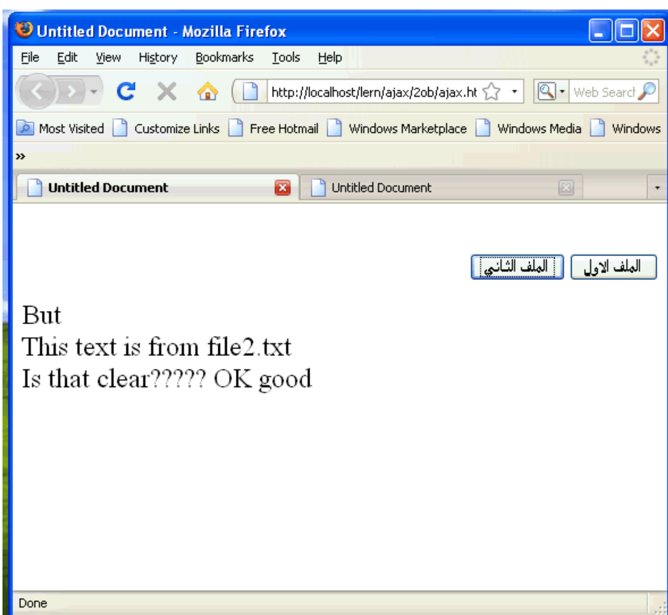
```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<script type="text/javascript">
var ajax1 = false;
var ajax2 = false;
if(window.ActiveXObject)
{
ajax1 = new ActiveXObject("Microsoft.XMLHTTP");
ajax2 = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax1 = new XMLHttpRequest();
ajax2 = new XMLHttpRequest();
}
//_______________
function getData1(dataSourse)
{
if(ajax1)
{
ajax1.open("GET",dataSourse);
ajax1.onreadystatechange = function()
{
if(ajax1.readyState == 4 && ajax1.status == 200)
{
document.getElementById("div1").innerHTML =
ajax1.responseText;
}
}
ajax1.send(null);
}
}
//__________________________________
//
//*****************
function getData2(dataSourse)
{
if(ajax2)
{
ajax2.open("GET",dataSourse);
ajax2.onreadystatechange = function()
{
if(ajax2.readyState == 4 && ajax2.status == 200)
{
document.getElementById("div1").innerHTML =
ajax2.responseText;
}
}
ajax2.send(null);
}

}
//*****************
</script>
<body>
```

```
<p> </p>
<p>
<form>
<input type="button" onClick="getData1('file1.txt')" value=" الملف الاول">
<input type="button" onClick="getData2('file2.txt')" value=" الملف الثاني">
</form>
</p>
<div dir="ltr" style="font-size:24px;" id="div1"></div>
</body>
</html>
```

كما لاحظت عرفنا متغيرين ajax1 و ajax2 و عرفنا ايضا دالتين getData1 و الدالة getData2 فهذا مفيد لتفادي ارباك السكربت و هذه هي النتيجة في المتصفح



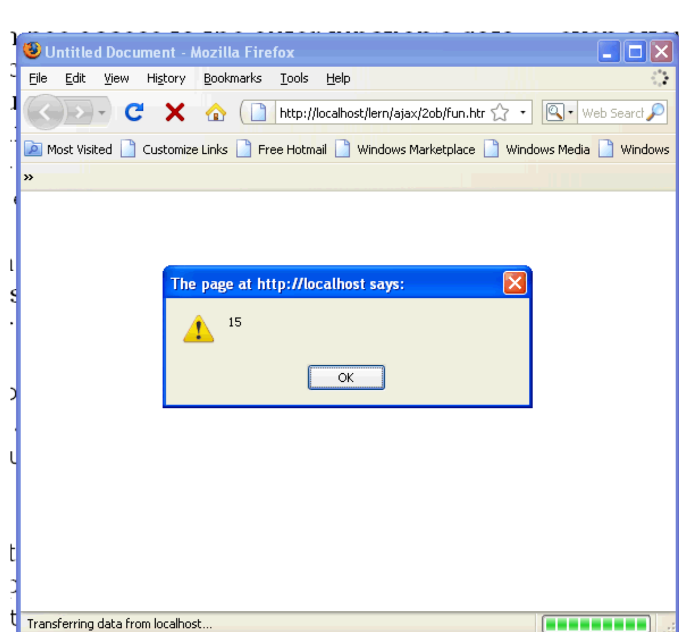


رائع جدا لكن الكود السابق يسمح لك بانشاء كائنين فقط XMLHttpRequest , و ماذا لو ضغط المستخدم على نفس الزر عدة مرات و بسرعة في كل مرة سيطلب طلب جديد من السيرفر قبل ان يكتمل تحميل الطلب القديم و يكون الحل هنا بانشاء فنكشن داخلية.

الفنكشن الداخلية inner function

و هي دالة عادية نحن نعرفها داخل دالة اخرى و نسميها كما نريد :

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript">
function outer(data)
{
var num1 = data;
//++++++++++++++++++++++++
function inner(num2)
{
alert(num1+num2);
}
//++++++++++++++++++++++++
inner(5);
}
</script>
</head>
<body onLoad="outer(10)" >
</body>
</html>
```



طبعا الفنكشن الداخلية لديها الصلاحيات لكي تصل الى المعامل الذي مررناه للدالة الخارجية .

هل تعلم انك في كل مرة تستدعي الفنكشن outer تصنع نسخة جديدة من هذه الفنكشن outer و بالتالي يتم انشاء قيمة جديدة من المتغير num1 الذي ستتعامل معه الفنكشن الداخلية inner بدورها و نفس الشيئ بالنسبة للفنكشن getData حيث سيتم انشاء نسخة جديدة منها تحوي كائن جديد XMLHttpRequest في كل مرة يتم استدعائها و ستكون الفنكشن الداخلية او التي كنا نسميها المجهولة لانها بدون اسم ستكون قادرة على الوصول الى هذا الكائن الجديد :

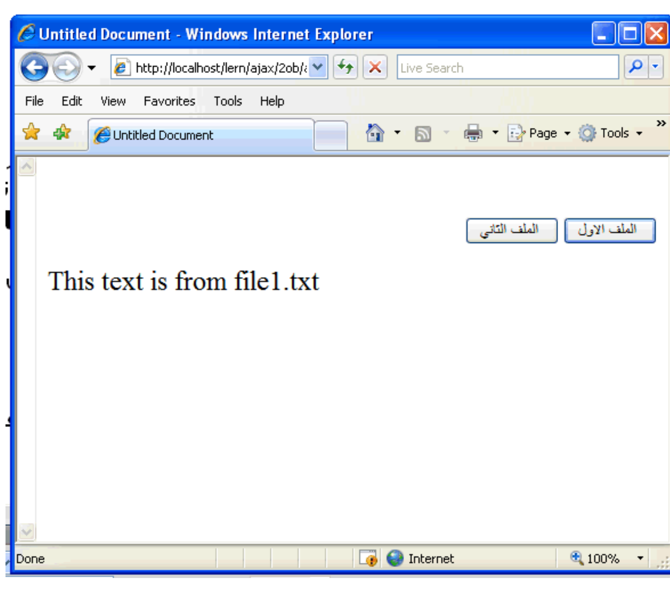
```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<script type="text/javascript">
//____________________________________
function getData(dataSourse)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if(ajax)
{
ajax.open("GET",dataSourse);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
document.getElementById("div1").innerHTML = ajax.responseText;
delete ajax;
ajax = null;
}
}
ajax.send(null);
}
}
//____________________________________
</script>
<body>
<p> </p>
<p>
<form>
<input type="button" onClick="getData('file1.txt')" value=" الملف الاول">
<input type="button" onClick="getData('file2.txt')" value=" الملف الثاني">
</form>
</p>
<div dir="ltr" style="font-size:24px;" id="div1"></div>
</body>
</html>
```

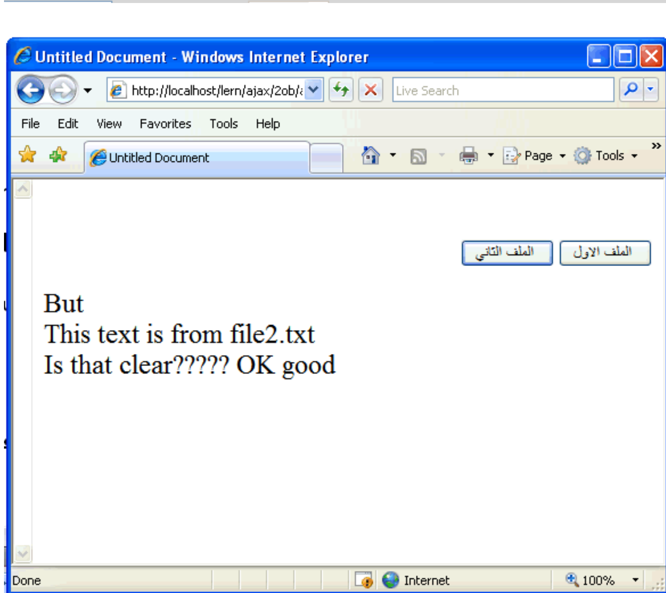
الجديد بالكود السابق اننا عرفنا الكائن XMLHttpRequest داخل الفنكشن **getData** و ليس خارجها و ذلك لكي يتم انشاء نسخة جديدة من الفنكشن getData في كل مرة يتم استدعائها و بالتالي نقدم للفنكشن الداخلية كائن XMLHttpRequest جديد يعني باختصار انت بذلك تصنع **XMLHttpRequest** جديد لكل طلبية

و لا مانع ان تحرر الذاكرة من الكائن كخطوة احترافية :

delete ajax;
ajax = null;

طبعا لن تتغير النتيجة الفرق الوحيد هو انك تخصص كائن جديد لكل طلب من السيرفر :





الفصل الرابع Frameworks

نحن تعودنا ان نبني سكربت الاجاكس من تحت الصفر و لكن هناك طريقة توفر عليك الكثير من الوقت و الجهد و هي ان تستخدم اجاكس Frameworks و هي عبارة عن ملفات جافا سكريبت تحوي مكتبات جاهزة من الدوال فقط قم بتضمين ملفات الجافا سكريبت ضمن برنامجك ثم قم باستدعاء الفنكشن التي تريد

ديناميك HTML

لا بد و انك سمعت سمعت عن DHTML و هو تغيير عناصر في صفحة الويب بدون اي تحديث للصفحة الان اريد ان تفكر و تحاول ان تجد طريقة لعرض صورة متغيرة اي تتغير هذه الصورة بدون اعادة تحميل الصفحة هذا سهل جدا باستخدام تقنية الاجاكس و خاصة انك مبرمج اجاكس محترف لن يصعب عليك ذلك

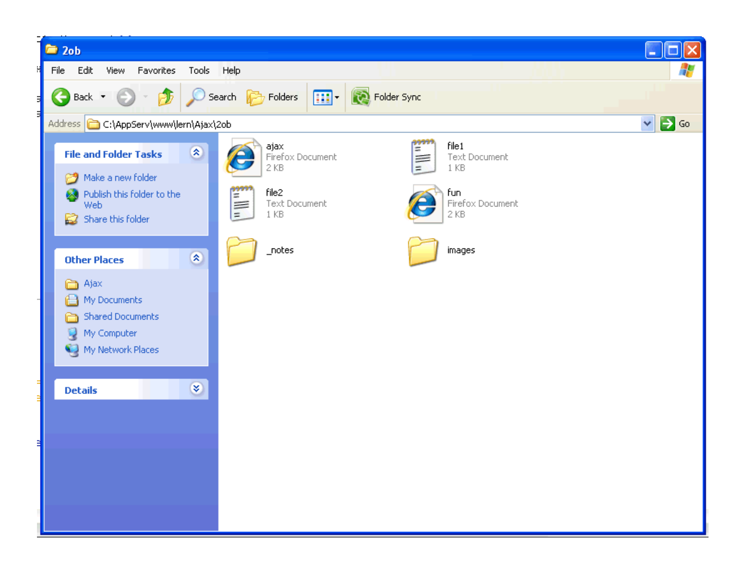
حسنا لا بد من استخدام الكائن XMLHttpRequest و لكن هذا الكائن يسمح لنا بجلب نص او اكس ام ال و لا يسمح لنا بسحب صور من السيرفر و الحل ببساطة ان نخزن رابط الصورة على ملف نصي ثم نحمل هذا النص و نضعه في عنصر img فيقوم المتصفح بتحميل الصورة . اريد ان تحاول ان تصنع السكربت بنفسك قبل ان تقرأ الكود :

```
<html dir="rtl">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>DHTML</title>
</head>
<script type="text/javascript">
//____________________________________
function getData(dataSourse)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if(ajax)
{
ajax.open("GET",dataSourse);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
var src = ajax.responseText
document.getElementById("div1").innerHTML = "<img src
='"+src+"' width=200>" ;
delete ajax;
ajax = null;
}
}
ajax.send(null);
}
}
//____________________________________
</script>
<body>
<table width="400" border="0" align="center" cellpadding="3"
cellspacing="3">
<tr>
<td width="195" align="center"><h3
onMouseOver="getData('file2.txt')">منتدى الكيمياء</h3></td>
<td width="178" align="center"><h3
onMouseOver="getData('file1.txt')">منتدى صور الطبيعة</h3></td>
</tr>
<tr align="center">
<td height="179" colspan="2"><div id="div1"></div></td>
```

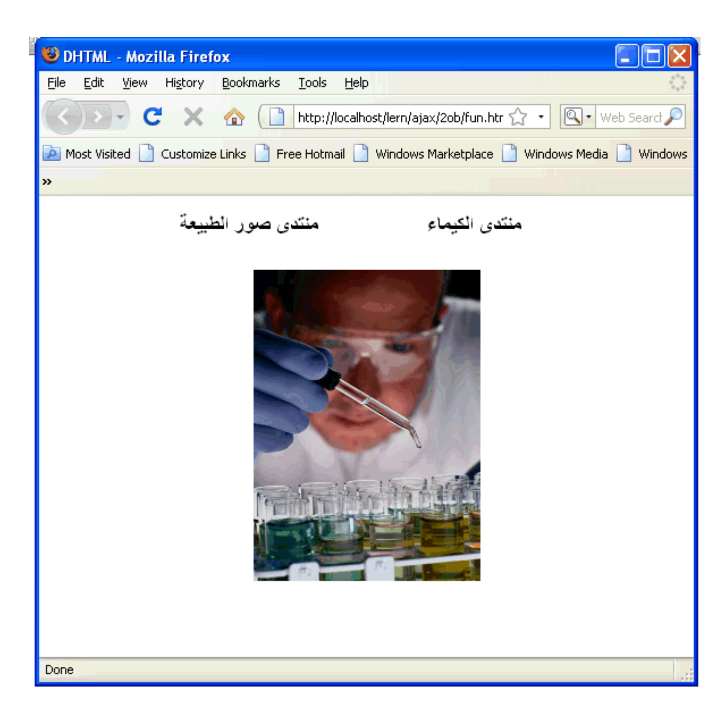


```
</tr>
</table>
</body>
</html>
```

شرط ان تضع مجلد الصور و الملفات النصية و ملف الهوتميل بنفس المجلد :



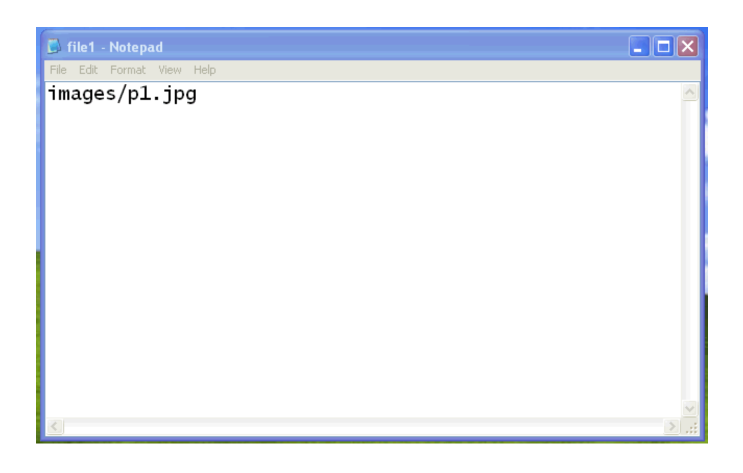
عند مرور الماوس فوق جملة منتدى الكيمياء تظهر صورة مناسبة



اما اذا مر مؤشر الماوس فوق منتدى صور الطبيعة :



و لاتنسى ان تكتب الرابط الصحيح في الملف النصي:



هيا لنبني Ajax Frameworks

كما اسلفنا استخدام Ajax Frameworks يوفر عليك الكثير من الوقت و الجهد و ما هي الا ملفات جافا سكريبت نقوم بتضمينها بصفحتنا فلنبدا ببناء ملف جافا سكريبت سنحفظه بالامتداد js :

```
// JavaScript Document
function getText(url,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if (ajax)
{
ajax.open("get",url);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
callback(ajax.responseText);
delete ajax;
ajax = null;
}
}
ajax.send(null);
}
}
```

احفظه باسم **ayman.js** و ليس بالامتداد **.html** الجديد عليك هو اننا مررنا للفنكشن getText المعامل الباراميتر الاول **url** و ثم الباراميتر الثاني **callback**:

```
function getText(url,callback)
```

قد تتسائل ماذا تعني **callback** ؟ نقصد بها اسم الفنكشن التي ستقوم بمعالجة النص القادم من السيرفر طبعا تعريف هذه الفنكشن سيكون بملف الاتش تي ام ال :

```
callback(ajax.responseText);
```

و الان لننتقل الى ملف get.html:

```
<html>
<head>
<style type="text/css">
#divID
{
background-color:#900;
font-size:24px;
color:#FF0;
width:300px;
overflow:auto;
}
</style>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript" src="ayman.js"></script>
<script type="text/javascript">
function message(text)
{
document.getElementById("divId").innerHTML = text;
}
//---------------------
function show(str)
{
document.getElementById("divId").innerHTML = str;
}
</script>
</head>
<body>
<form>
<input type="button" value=" رسالة ترحيب"
onClick="getText('file1.txt',message)">
<input type="button" value=" تاريخ الماي اس كيو ال"
onClick="getText('file2.txt',show)">
</form><br>
<div id="divId" align="center"></div>
</body>
</html>
```

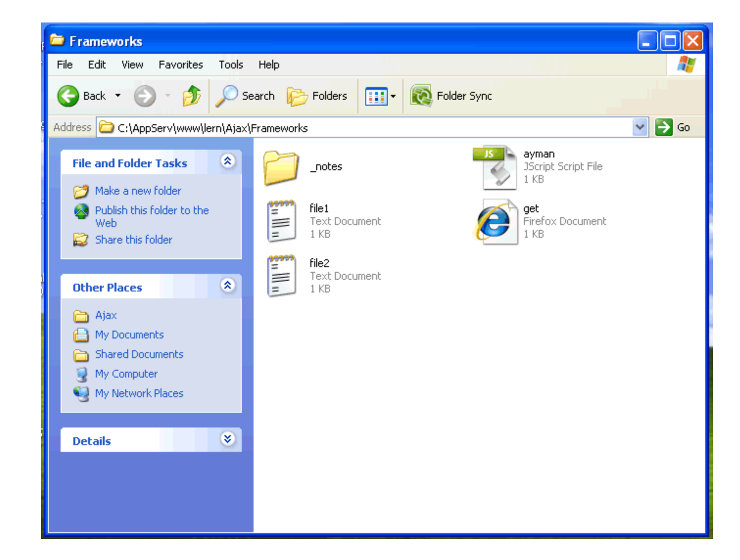
اولا قمنا بتضمين ملف الجافا سكريبت ضمن عنصر **script**
و لكن اياك ان تعرف الفنكشن داخل هذا السطر :

```
<script type="text/javascript" src="ayman.js">اقصد لا تضع شيئأ هنا
</script>
```

و عندما تريد ان تعرف الفنكشن افتح عنصر **script** جديد هكذا :

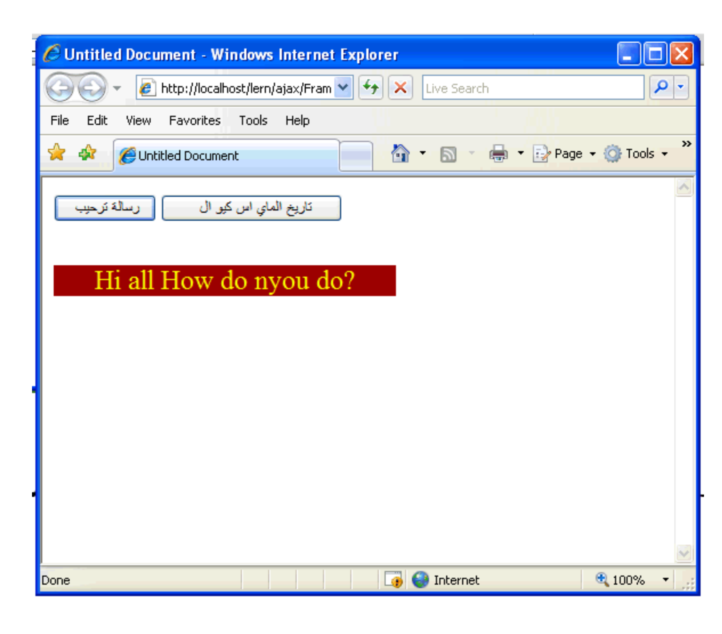
```
<script type="text/javascript">
function message(text)
{
document.getElementById("divId").innerHTML = text;
}
//---------------------
function show(str)
{
document.getElementById("divId").innerHTML = str;
}
```

ثم انشأنا فورم و ازرار و لاحظ ان كل زر يستدعي نفس الفنكشن **getText** ثم يمرر لها فنكشن مختلفة مثلا **show** او **message** و لا حظ ايضا ان الملفات النصية و الجميع بنفس المجلد :

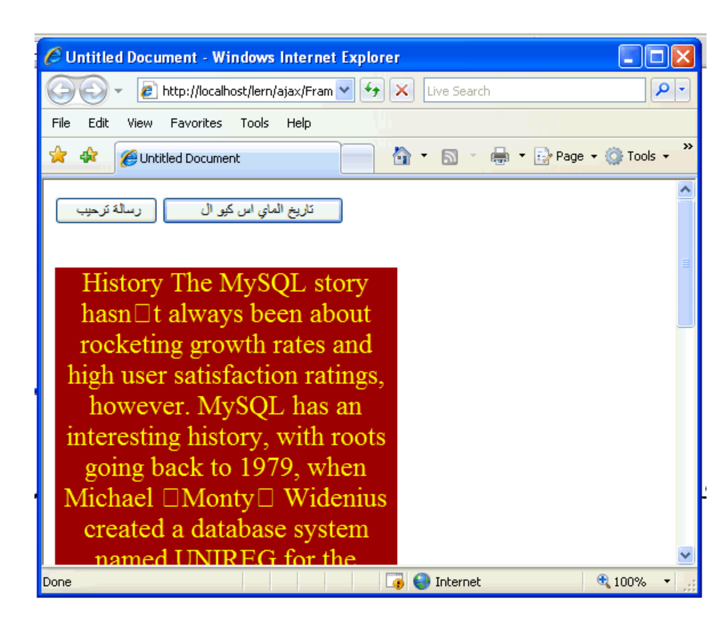


ملاحظة اخيرة احيانا الدريم ويفر يضيف تلقائيا **<label>** مع الفورم قم بحذفه فانت لا تحتاج الى عنصر <label>.

و الان لنجربه بالمتصفح:



اما اذا ضغطت على الزر الثاني:



get Xml

الان سنقوم بجلب محتويات ملف بي اتش بي يحوي بيانات اكس ام ال اولا اكتب هذا الكود و احفظه باسم script1.php :

```
<?php
header("Content-type: text/xml");
print "<?xml version='1.0' encoding='utf-8' ?>";
print "<alloptions>";
$arr = array("red","green","blue","black");
for($i=0;$i<count($arr);$i++)
{
print "<option>".$arr[$i]."</option>";
}
print "</alloptions>";
?>
```

و لدينا ملف اخر احفظه باسم script2.php و هذا هو الكود :

```
<?php
header("Content-type: text/xml");
print "<?xml version='1.0' encoding='utf-8' ?>";
print "<alloptions>";
$arr = array("yellow","pink");
for($i=0;$i<count($arr);$i++)
{
```

```
print "<option>".$arr[$i]."</option>";
}
print "</alloptions>";
?>
```

و الان لننتقل الى ملف الجافا سكريبت الموجود مسبقا لديك باسم ayman.js فقط نريد ان نضيف دالة جديدة getXml فيصبح الملف ayman.j هكذا :

```
// JavaScript Document
//************************
function getText(url,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if (ajax)
{
ajax.open("get",url);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
callback(ajax.responseText);
delete ajax;
ajax = null;
}
}
ajax.send(null);
}
}
//************************
// getXml
//++++++++++++++++++++++++++++++++++++++++
function getXml(url,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if (ajax)
{
ajax.open("get",url);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
xmlDocument = ajax.responseXML;
callback(xmlDocument);
}
}
ajax.send(null);
}
}
//++++++++++++++++++++++++++++++++++++++++
```

الفرق الوحيد هو اننا استخدمنا ajax.responseXML بدلا من الخاصيةresponseText لاننا نتعامل مع بيانات اكس ام ال و الان بقي علينا ملف الهوتميل get_xml.html :

```
<html>
<head>
<script type="text/javascript" src="ayman.js"> </script>
<script type="text/javascript" >
//--------------------------------------
function makeOptions(xml)
{
var optionsArr = xml.getElementsByTagName("option");
var selectControl = document.getElementById("selectId");
selectControl.options.length = 0;
for(i=0;i< optionsArr.length;i++)
{
selectControl.options[i] = new
Option(optionsArr[i].firstChild.data);
}
}
//--------------------------------------
//______________________________
function setColor()
{
var optionsArr = xmlDocument.getElementsByTagName("option");
index = document.getElementById("selectId").selectedIndex;
document.getElementById("div1").style.color =
optionsArr[index].firstChild.data;
}
//______________________________
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ayman for programming</title>
</head>
<body>
<form>
<select id="selectId" onChange="setColor()">
<option> choos</option>
</select>
<input type="button"
onClick="getXml('script1.php',makeOptions)" value="مجموعة الالوان الاولى">
<input type="button"
onClick="getXml('script2.php',makeOptions)" value="المجموعة الثانية">
</form>
<br>
<div id="div1">
<h1>I love Ajax </h1>
</div>
</body>
</html>
```

حيث تقوم الفنكشن makeOptions بمعالجة بيانات الاكس ام ال و تحويلها الى نصوص عادية عن طريق الخاصية firstChild.data و ثم نقوم بعرضها كقائمة اختيارات :

```
selectControl.options[i] = new Option(optionsArr[i].firstChild.data);
```

و لدينا ايضا الفنكشن setColor التي مهمتها تغيير لون النص حسب ما يرغب المستخدم و ذلك باستخدام الخاصيتين style.color بعد معرفة رقم الاندكس للعنصر الذي تم اختياره:

```
document.getElementById("div1").style.color =
optionsArr[index].firstChild.data;
```

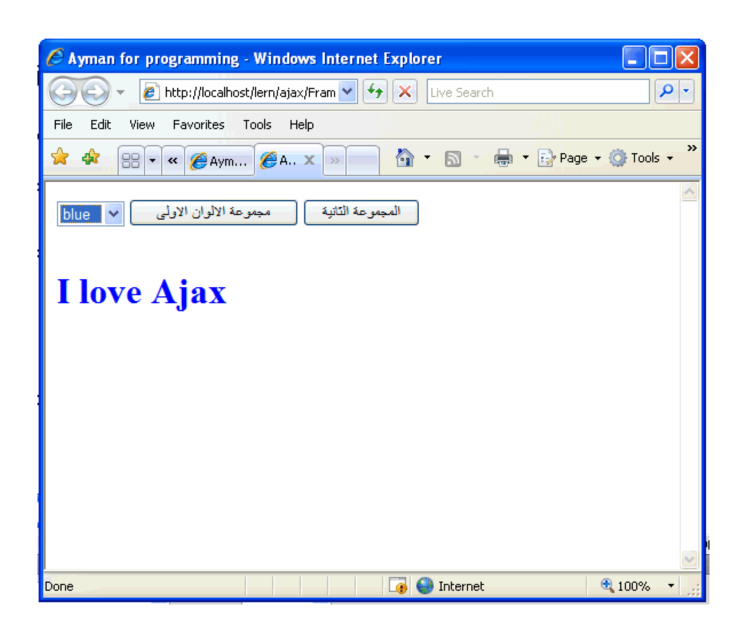
اما هذا السطر فنقوم بحذف محتويات القائمة لتعبئتها بعناصر جديدة:

```
;selectControl.options.length = 0
```

و اخيرا انشأنا زرين ليكون للمستخدم الخيار بانتقاء مجموعة الالوان المناسبة :

```
<input type="button" onClick="getXml('script1.php',makeOptions)"
value="مجموعة الالوان الاولى">
<input type="button" onClick="getXml('script2.php',makeOptions)"
value="المجموعة الثانية">
```

و الان لنستعرض برنامجنا بالمتصفح :



نجح البرنامج بفضل الله تعالى و بامكانك تغيير لون النص .

postText

الان سنضم فنكشن جديدة الى مكتبتنا الذهبية و هي الفنكشن postText طبعا لن نفتح ملف جديد و انما سنكتب بملفنا السابق ayman.js الكود التالي:

```
// postText
function postText(url,data,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if (ajax)
{
ajax.open("POST",url);
ajax.setRequestHeader('Content-Type','application/x-www-form-
urlencoded');
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
callback(ajax.responseText);
delete ajax;
ajax = null;
}
}
ajax.send(data);
}
}
//______________________________________
```

طبعا استخدمنا الخاصية اوبن لفتح الملف و مررنا لها القيمة بوست بلا من جيت :

```
ajax.open("POST",url);
```

ثم وضعنا قيمة للهيدر طبعا لا داعي ان نضع الهيدر مع جيت :

```
ajax.setRequestHeader('Content-Type','application/x-www-form-
urlencoded');
```

و اخيرا مررنا قيمة داتا للخاصية سيند عوضا عن نال التي نمررها عند استخدامنا جيت :

```
ajax.send(data);
```

و الان الى ملف البي اتش بي print.php بسيط جدا :

```
<?php
print "<h2> Welcome ".$_POST[user_name]."</h2>";
?>
```

بقي علينا ملف الهوتميل post.html الذي سيحوي دالة مهمتها عرض النص القادم من السيرفر :

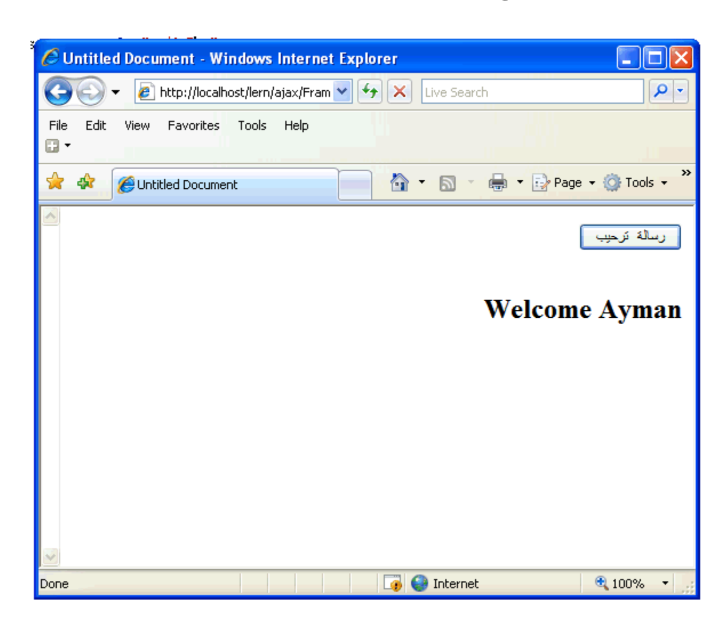
```
<html dir="rtl">
<head>
<script type="text/javascript" src="ayman.js"></script>
<script type="text/javascript">
function hi(text)
{
document.getElementById("div1").innerHTML = text;
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<form>
<input type="button" value="رسالة ترحيب"
onClick="postText('print.php','user_name=Ayman',hi)" >
</form>
<br>
<div id="div1"></div>
</body>
</html>
```

كما ترى الفنكشن postText نمرر لها ثلاث باراميترات :

- عنوان الملف
- البيانات التي نريد ارسالها للسيرفر داتا
- اسم الدالة التي ستعالج النص القادم من السيرفر

```
postText('print.php','user_name=Ayman',hi)
```

و الان لنجرب بالمتصفح:



postXml

سنضيف بعون الله الفنكشن postXml الى مكتبتنا الذهبية التي ترسل معلومة الى السيرفر مثلا رقم مجموعة الالوان ثم تستقبل بيانات اكس ام ال و توكل مهمة معالجة هذه البيانات و عرضها على شكل قائمة اختيارات الى الفنكشن makeOptions:

```
//****************
function postXml(url,data,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if(ajax)
{
ajax.open("post",url);
ajax.setRequestHeader('Content-Type','application/x-www-form-
urlencoded');
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
xmlDocument = ajax.responseXML;
callback(xmlDocument);
}
}
ajax.send(data);
}
}
//****************
```

طبعا الكود السابق ليس ملف مستقل و انما اضفه الى ملف الجافا سكريبت ayman.js

اول فرق بين هذا الكود و سابقه اننا استخدمنا الخاصية responseXML لاننا نستقبل بيانات اكس ام ال :

```
xmlDocument = ajax.responseXML;
callback(xmlDocument);
```

لننتقل الى ملف البي اتش بي colors.php:

```
<?php
extract($_POST);
if($group == 1)
{
$arr = array("yellow","pink");
}
else if($group == 2)
{
$arr = array("red","green","blue","black");
}
header("Content-type: text/xml");
print "<?xml version='1.0' encoding='utf-8' ?>";
print "<allOptions>";
for($i=0;$i< count($arr);$i++)
{
print "<option>".$arr[$i]."</option>";
}
print "</allOptions>";
?>
```

وضعنا شرط اذا كان المتغير جروب يحمل القيمة واحد هناك لونين مجموعة الالوان الاولى :

```
$arr = array("yellow","pink");
```

اما اذا كان المتغير جروب يحمل القيمة اثنين سنخزن بالمصفوفة اربع الوان :

```
$arr = array("red","green","blue","black");
```

لم يتبق علينا الا ملف الهوتميل post_xml.html :

```
<html>
<head>
<script type="text/javascript" src="ayman.js"></script>
<script type="text/javascript">
function makeOptions(xml)
{
var optionsArr = xml.getElementsByTagName("option");
var i;
var selectControl = document.getElementById("selectId");
selectControl.options.length = 0;
for(i=0;i< optionsArr.length;i++)
{
selectControl.options[i] = new
Option(optionsArr[i].firstChild.data);
}
}
//***********
function setOption()
{
var optionsArr = xmlDocument.getElementsByTagName("option");
var index = document.getElementById("selectId").selectedIndex;
document.getElementById("div1").style.color =
optionsArr[index].firstChild.data;
}
//**********
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<form>
<select id="selectId" onChange="setOption()">
<option>Choos one</option>
</select>
<input type="button" value="مجموعة الالوان الاولى"
onClick="postXml('colors.php','group=1', makeOptions)"/>
<input type="button" value="مجموعة الالوان الثانية"
onClick="postXml('colors.php','group=2', makeOptions)"/>
</form><br>
<div id="div1">
<h1>Ayman for programming</h1>
</div>
</body>
</html>
```

اولا عرفنا الدالة التي مهمتها معالجة البيانات القادمة من السيرفر :

```
function makeOptions(xml)
{
var optionsArr = xml.getElementsByTagName("option");
}
```

ثم تقوم هذه الدالة بحذف جميع عناصر القائمة:

```
selectControl.options.length = 0;
```

ثم تقوم ببناء قائمة جديدة حسب ما اختار المستخدم :

```
for(i=0;i< optionsArr.length;i++)
{
selectControl.options[i] = new Option(optionsArr[i].firstChild.data);
}
```

اخيرا لدينا الفنكشن setOption التي مهمتها تغغير اللون حسب رغبة المستخدم باستخدام الخاصيتين style.color :

```
function setOption()
{
var optionsArr = xmlDocument.getElementsByTagName("option");
var index = document.getElementById("selectId").selectedIndex;
document.getElementById("div1").style.color =
optionsArr[index].firstChild.data;
}
```

و طبعا انشأنا زرين لكي نسمح للمستخدم بانتقاء مجموعة الالوان المناسبة :

```
<input type="button" value="مجموعة الالوان الاولى"
onClick="postXml('colors.php','group=1', makeOptions)"/>
<input type="button" value="مجموعة الالوان الثانية"
onClick="postXml('colors.php','group=2', makeOptions)"/>
```

و اخيرا اريد ان اضع لك الملف الذهبي الذي اسميناه Framework ayman.js ساضعه لك كاملا لكي تستفيدة منه بتطبيقاتك فقط قم بتضمين الملف دون الحاجة ان تعيد كتابة تلك الدوال كل مرة:

```
// JavaScript Document
//************************
function getText(url,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if (ajax)
{
ajax.open("get",url);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
callback(ajax.responseText);
delete ajax;
ajax = null;
}
}
ajax.send(null);
}
}
//************************
// getXml
//++++++++++++++++++++++++++++++++++++++++
function getXml(url,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if (ajax)
{
ajax.open("get",url);
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
xmlDocument = ajax.responseXML;
callback(xmlDocument);
}
}
ajax.send(null);
}
}
//________________________________________
// postText
function postText(url,data,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if (ajax)
{
ajax.open("POST",url);
ajax.setRequestHeader('Content-Type','application/x-www-form-
urlencoded');
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
callback(ajax.responseText);
delete ajax;
ajax = null;
}
}
ajax.send(data);
}
}
//________________________________________
//postXml
//****************
function postXml(url,data,callback)
{
var ajax = false;
if(window.ActiveXObject)
{
ajax = new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest)
{
ajax = new XMLHttpRequest();
}
if(ajax)
{
ajax.open("post",url);
ajax.setRequestHeader('Content-Type','application/x-www-form-
urlencoded');
ajax.onreadystatechange = function()
{
if(ajax.readyState == 4 && ajax.status == 200)
{
xmlDocument = ajax.responseXML;
callback(xmlDocument);
}
}
ajax.send(data);
}
}
//****************
```